SAE Editor User Guide

V 0.9.9 (30)

// SDMX Structure Artefact Editor (SAE). An SDMX-ML Structure 2.0 Editor.
//   Copyright (C) 2009-2012 Philippe Prati
//
//   This program is free software: you can redistribute it and/or modify
//   it under the terms of the GNU General Public License as published by
//   the Free Software Foundation, either version 3 of the License, or
//   (at your option) any later version.
//
//   This program is distributed in the hope that it will be useful,
//   but WITHOUT ANY WARRANTY; without even the implied warranty of
//   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
//   GNU General Public License for more details.
//
//   You should have received a copy of the GNU General Public License
//   along with this program. If not, see <http://www.gnu.org/licenses/>.
//
//   Contact: sdmx@nextsoft.ch   Download: http://www.sdmx.ch

Acknowledgement

Excel Support:

Free library to generate Excel Workbooks using XML
http://www.carlosag.net/Tools/ExcelXmlWriter/

Fast Excel file reader with basic functionality by Messir Alex
http://www.codeproject.com/KB/office/Excel_DataReader.aspx

PDF Support:

iTextSharp for PDF generation as GNU Library
http://sourceforge.net/projects/itextsharp/

RTF Support:

NET RTF Writer Library in C# as GNU Library
http://sourceforge.net/projects/netrtfwriter/

CSV fast Reader MIT License
http://www.codeproject.com/KB/database/CsvReader.aspx

Special thanks to Jean-François Fracheboud, Petko Yanev, Olivier Rütti and Michel Meyer for their suggestions and support.

TABLE OF CONTENTS

1 OVERVIEW 6

2 OPERATING MODES 7

2.1 Standalone mode 7

2.2 Repository basic mode 7

2.3 Repository with ownership management 7

2.4 Multiple independent repositories 8

2.5 Multiplexed Repositories 8

3 SAE GUI 9

3.1 Areas 9

3.2 Menu 10
3.2.1 File Menu 10
3.2.2 Repository Menu 10
3.2.3 Find Menu 11
3.2.4 Import Menu 11
3.2.5 Export Menu 11
3.2.6 Categories Menu 11
3.2.7 Data Menu 11
3.2.8 Utilities Menu 11
3.2.9 Help Menu 12

3.3 Toolbar 12
3.3.1 File Menu shortcuts 12
3.3.2 Repository shortcuts 12
3.3.3 Active language 12
3.3.4 Expands / Collapses selected tree item 13
3.3.5 Navigation functions 13

3.4 Tree view 14
3.4.1 Quick search 14
3.4.2 Popup Menu 15

3.5 Annotations View 16
3.5.1 Functionalities 16

3.6 Status bar 16
3.6.1 Tree Childs count 16
3.6.2 Registry connection status 16

3.7 Detail view 17
3.7.1 Read Only 17

4 INTERNATIONALIZATION SUPPORT 18

4.1 Language set definition 18

4.2 Multi Language edition 18

5 MENU REPOSITORY 20

5.1 Introduction 20

5.2 Usage from SAE Editor 20
5.2.1 Menu Repository – Define repository WS source… 20
5.2.2 Menu Repository – Get Artefact 21
5.2.3 Menu Repository – Reserve Artefact (check out) 22
5.2.4 Menu Repository – Submit Artefact (check in) 23
5.2.5 Menu Repository – Submit aggregated Artefacts to Repository 24
5.2.6 Menu Repository – Update Texts of Finalized Artefact 25
5.2.7 Menu Repository – Import local Image copy 25

6 OTHER MENUS 25

6.1 Menu Find 25
6.1.1 Locate References... 25
6.1.2 Find Annotations... 27

6.2 Menu Import / Export 29

6.3 Menu Categories 29
6.3.1 Sub menus definition: 30

6.4 Menu Data 31
6.4.1 SQL... 31
6.4.2 CSV… 31

6.5 Menu Utilities 33
6.5.1 Copy Artefact Ref. 33
6.5.2 DSD Wizard... 34

7 ARTEFACTS DETAIL VIEW 36

7.1 Organization Scheme 36
7.1.1 Artefact detail 36
7.1.2 Agency, Data Provider and Data Consumer detail 36
7.1.3 Organisation tab 37
7.1.4 Contacts Detail tab 37

7.2 Code Lists 39
7.2.1 Artefact detail 39
7.2.2 Code detail 39
7.2.3 Code hierarchy management 40

7.3 Concept Scheme 41
7.3.1 Artefact detail 41
7.3.2 Stand Alone concepts 41
7.3.3 Concept detail 42
7.3.4 Concept hierarchy management 42

7.4 DSD 43
7.4.1 Artefact detail 43
7.4.2 Dimensions 44
7.4.3 Measures 45
7.4.4 Attributes 46

7.5 Category scheme **47**
7.5.1 Artefact detail 47
7.5.2 Category detail 47
7.5.3 Category hierarchy management 48

7.6 Hierarchical Code List **49**
7.6.1 Artefact detail 49
7.6.2 Code lists reference 50
7.6.3 Information Tab details 50
7.6.4 Hierarchy Tab details 50
7.6.5 Level Tab details 52

7.7 MSD **54**
7.7.1 Artefact detail 54
7.7.2 Target Identifiers 54
7.7.3 Report 56

7.8 Metadata set **57**
7.8.1 Artefact detail 57
7.8.2 Artefact row detail 58
7.8.3 Row Attributes values 59
7.8.4 report of metadata embedding HTML. 60
7.8.5 Sample report of metadata set row. 61

7.9 Data flow and Metadata flow **62**
7.9.1 Artefact detail 62
7.9.2 Reference 62

7.10 Reporting Taxonomy **63**
7.10.1 Artefact detail 63

7.11 Processes **64**
7.11.1 Artefact detail 64
7.11.2 Process Step detail 64
7.11.3 Process Step Transitions 65
7.11.4 Functionalities 65

7.12 Structure Sets **66**
7.12.1 Artefact detail 66
7.12.2 Maps detail 66

8 EXCEL **68**

8.1 Excel Import dialog **68**

8.2 Loading an Excel Worksheet **69**

8.3 Mapping property editor **70**
8.3.1 Internationalized text mapping 70
8.3.2 Hierarchy mapping 70
8.3.3 Header rows 71
8.3.4 Artefact main attributes 71

8.4 Artefact creation **71**
8.4.1 Function create 71
8.4.2 Function update text 72
8.4.3 Samples 72

8.5 Excel Export **72**

9 PLUG-IN 73

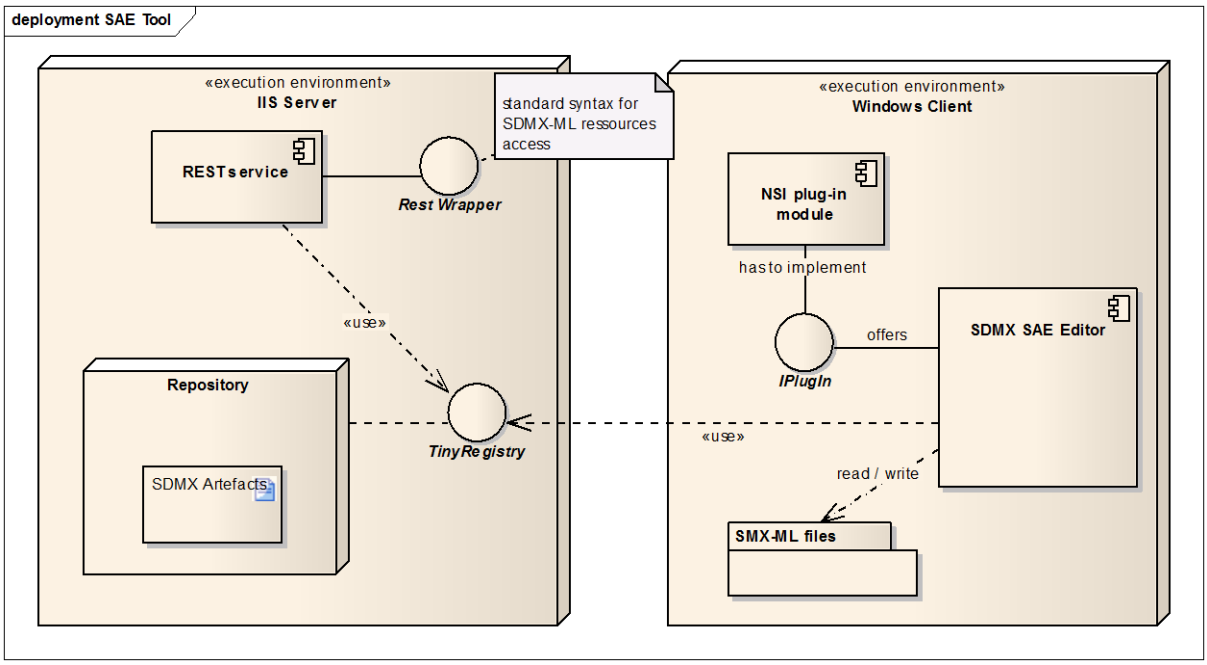
9.1 Plug-in installation 73
9.1.1 Required annotations 73

9.2 Plug-in demo 74

9.3 Interface definition 75

1 Overview

This application is created to manage statistical metadata through SDMX-ML documents. It is currently used in our NSI. It was developed in my free time and is enhanced according to our needs.
The application is composed of the SAE SDMX-ML editor (Windows Form .NET), of an artefact repository (Web service) and of a Restful service for easy consultation.



Producing SDMX document through Database and XML – XSLT transformation is quite easy, so the goal of this application is mainly to be an integration and navigation tool for non IT people. This application is oriented on SDMX ML V.2.0 Structure Artefacts. The idea is to hide XML complexity and to illustrate how statistical metadata are integrated in SDMX ML Framework.

The application adds artefacts in project by reading different SDMX files documents or by creating them from scratch. It can then submit or retrieve artefacts through the repository. When an artefact is fetched from the repository, the direct dependencies are also fetched (For example: Code Lists and Concepts Schemes referred by the requested DSD)

The user navigates between the artefacts of the project by using defined references or by manual selection.

The application allows creating complex SDMX documents files with different kinds of artefacts. On the other side, the repository holds single artefact documents.

A plug-in mechanism is offered in order to add NSI specific functionalities inside the editor.

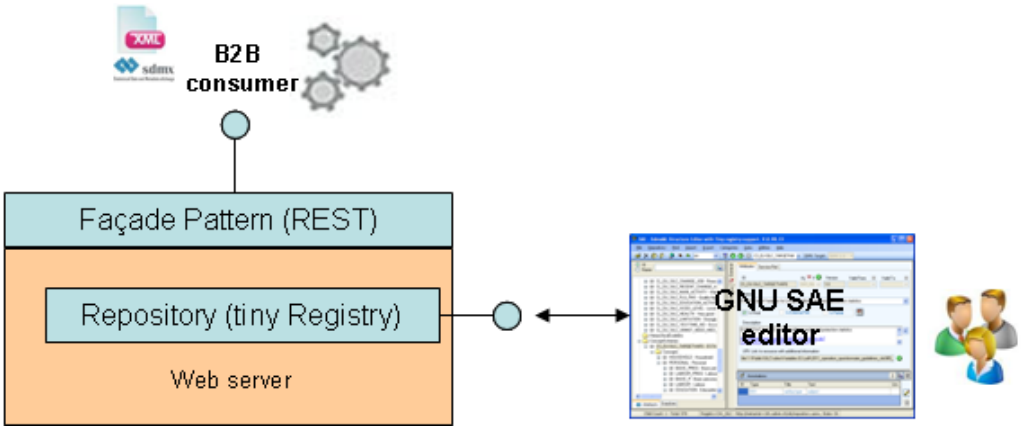
A Restful service wrapper allows easy consultation of repository content for others NSI tools

2 Operating modes

2.1 Standalone mode

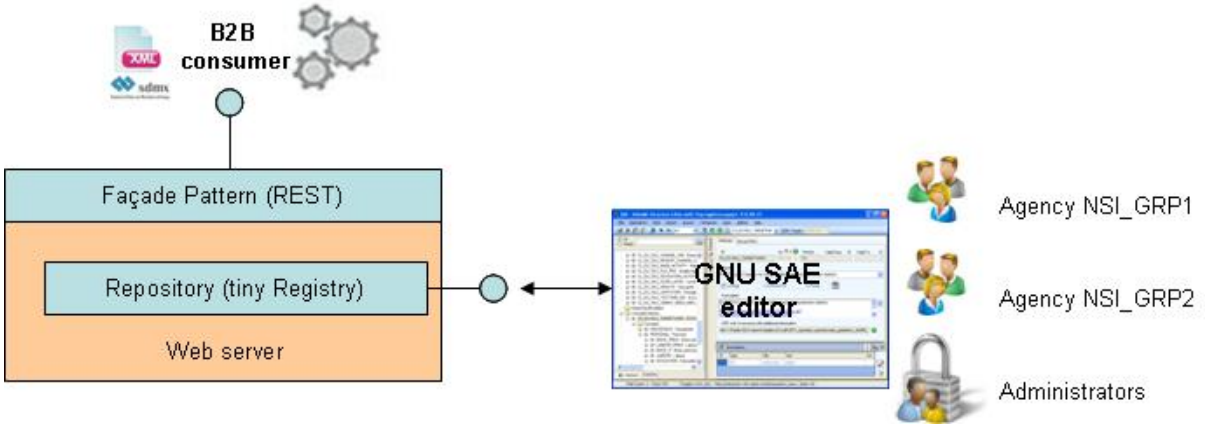
In this mode the metadata are managed only through XML files based on SDMX-ML model.

2.2 Repository basic mode



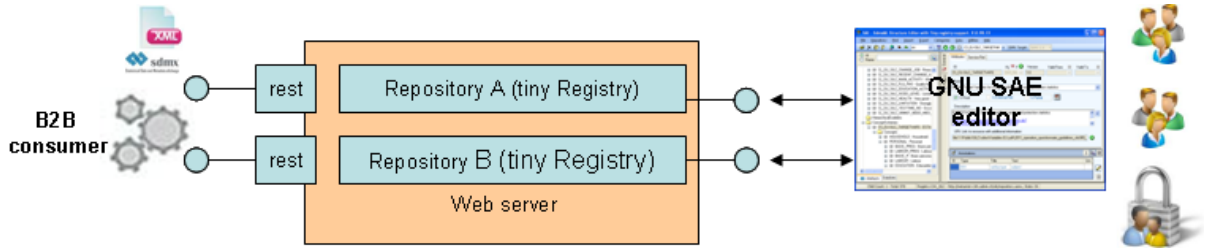
In this mode the metadata are managed through a repository web service. The basic mode allows every one to manage every metadata of the repository.

2.3 Repository with ownership management



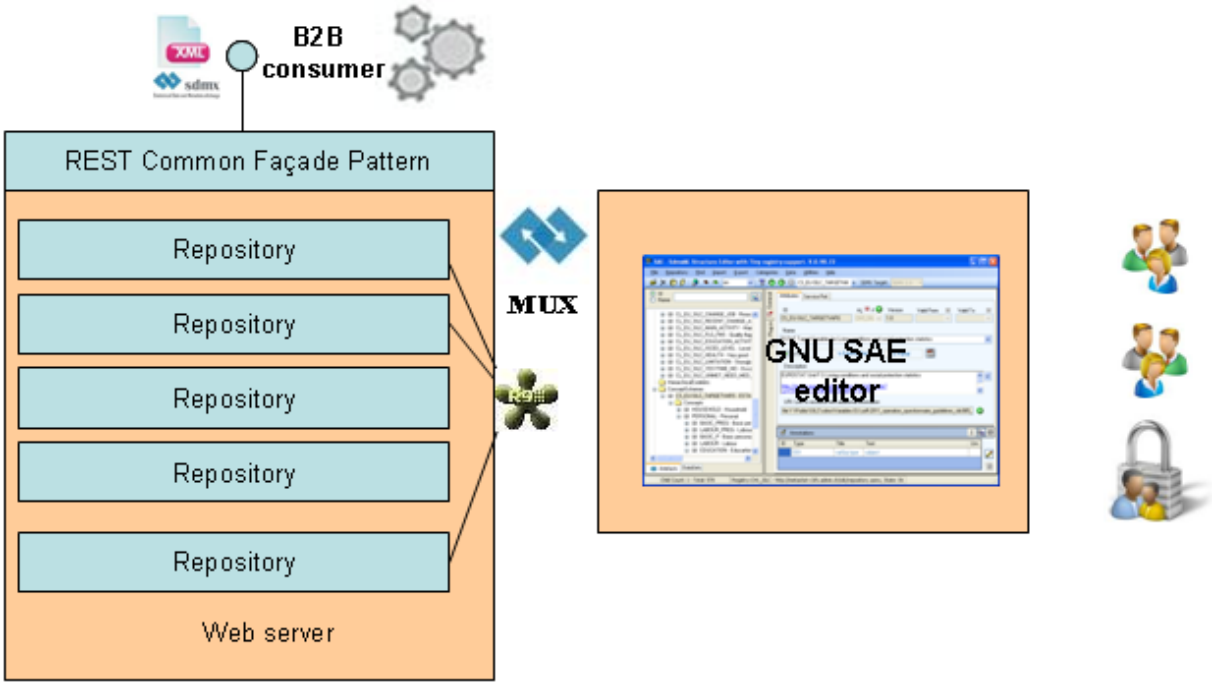
In this mode the metadata are managed through a repository web service. Owner groups (Agencies) are defined with authorized users according to windows identity. A group can only manage metadata of its agency. System administrators can manage everything.

2.4 Multiple independent repositories



In this mode the metadata are managed through a set of repositories web service. Each repository contains a coherent set of metadata (all references located in same repository). They can be managed with or without ownership support. The selection of working registry is done manually in the editor by the users.

2.5 Multiplexed Repositories



The set of repositories constitutes a single pool of coherent metadata. References can cross repository border. Agency groups are assigned to one of the registry. Ownership management is required, combined with group to registries assignation.

3 SAE GUI

3.1 Areas

The application screen is divided in the following areas. The functionality of each area is described in the next paragraph.

<table>
<tr><td colspan="2">Menu Bar</td></tr>
<tr><td colspan="2">Main Toolbar</td></tr>
<tr><td>Quick Search in tree</td><td rowspan="2">Detail View of selected element in tree view
Alternate view is plug-in view</td></tr>
<tr><td rowspan="2">Artefacts Tree view

Popup menu</td></tr>
<tr><td>Annotations View</td></tr>
<tr><td colspan="2">Status Bar</td></tr>
</table>

3.2 Menu

3.2.1 File Menu

Open New

Opens an SDMX structure or (meta) dataset document and clears previous memory content. The user is prompted to select the document that should be loaded.

Append SDMX File

Opens an SDMX structure or (meta) dataset document and appends it to the artefacts already in memory. If the document contains artefacts that, according to their identification, are already in memory, these artefacts are ignored. The user is prompted to select the document that should be loaded.

Save All

It saves all SDMX structure artefacts currently in memory in a unique SDMX document. For its activation, uses tree popup or menu “Save selected Artefact” in order to save a specific artefact or a typed collection of artefacts.

Note: Dataset and generic MetadataSet in memory are not affected by this saving. They can also be saved through tree popup menu functionality.

Save selected Artefact

Writes the artefact selected in tree as SDMX-ML document. If an artefact collection is currently selected in tree, it saves the complete collection.

Clear All

This command removes all artefacts currently in the project memory.

3.2.2 Repository Menu

All commands under this menu domain require a valid link with the “TinyRegistry” web services

Define Repository WS source

Provides a dialog box where an available instance of TinyRegistry source can be selected to work with.

Get Artefact

This command provides a dialog box giving access to the Repository service. From there all available artefacts can be searched and loaded.

Reserves Artefact (Check out)

Reserves selected artefact in repository for unique usage of Windows Identity user.

Submit Artefact (Check in)

Submit an update of selected artefact in repository or cancel previous reservation.

Submit Aggregated Artefacts

Submit to the repository the whole memory project as an aggregated artefacts document.

Import local Image copy

This command raises a dialog box allowing user to parameter a copy of the whole content of the repository in a given local directory.

3.2.3 Find Menu

Locate References

This function locates the references of the code list or concept selected in main tree. A non-modal dialog box is displayed with all references found in project memory (in Concept Scheme, DSD or MSD). It is then possible to activate automatically the detailed view of the artefact item holding the selected reference.

Replacement of reference(s) is also available.

Find in Annotations

This menu provides a dialog box for search criteria specification. It is then possible to activate automatically the detailed view of the artefact item holding the selected annotation.

3.2.4 Import Menu

Excel

Provides a dialog box where the artefact type and the fields mapping can be specified. The Excel content can then be imported as an SDMX artefact.

SDMX-ML by URN

This command provides a dialog box to specify the URI locator of a SDMX-ML document to append in project memory.

3.2.5 Export Menu

Excel

Exports selected artefact as an Excel document.

CSV comma-separated-values

Exports selected code list as an CSV document. User can choose separator in submenu commands.

3.2.6 Categories Menu

If configured, it provides a set of entry points for navigation from Repository. Full ID of artefacts to load must be defined in the configuration file of the application.

3.2.7 Data Menu

SQL…

Experimental, Creates from a selected DSD a set of SQL commands for construction of Oracle tables (dim and facts).

CSV…

Read CSV data file and bind it with DSD metadata, a validity control is available. It can operate on huge file through a chunking mechanism.

3.2.8 Utilities Menu

Copy Artefact Ref.

This function creates in the MS-Window clipboard the urn reference of the artefact or its sub item selected in the tree view. The urn can be pasted in concept or DSD edition for code list or concept reference, it can also be copied as text or as annotation urn.

Sample of urn:

urn:sdmx:org.sdmx.infomodel.codelist.Code=ESTAT:CL_GEO[1.1].IT

DSD Wizard
This function allows quick creation of a DSD skeleton. It creates also associated codelist for Cross sectional measures. See: 5.2.3

3.2.9 Help Menu

User Manual…

This command provides access to this document.

Documentation SMS

Provides an access to NSI documents located in the SMS sub directory of the editor

Documentation SDMX

Provides an access to SDMX documents located in the SDMX sub directory of the editor

Send feed back

It raises your mail system to send a feedback or question about this tool.

About

It raises the About dialog box.

3.3 Toolbar

3.3.1 File Menu shortcuts

	Append SDMX File
	Write All artefacts loaded in tree view as a unique SDMX Document
	Save selected Artefact as a SDMX Document

3.3.2 Repository shortcuts

	Load dialog for selecting artefact from repository
	Reserve (lock) selected artefact in repository
	Submit selected artefact to repository

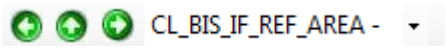
3.3.3 Active language

The combo box allows selecting the language to use for displaying internationalized information. A dynamic refresh is executed on displayed texts.

3.3.4 Expands / Collapses selected tree item

This function allows a full expand or collapse of the selected item in tree. Combined with navigation functions, it allows a quick consultation on artefact items information.

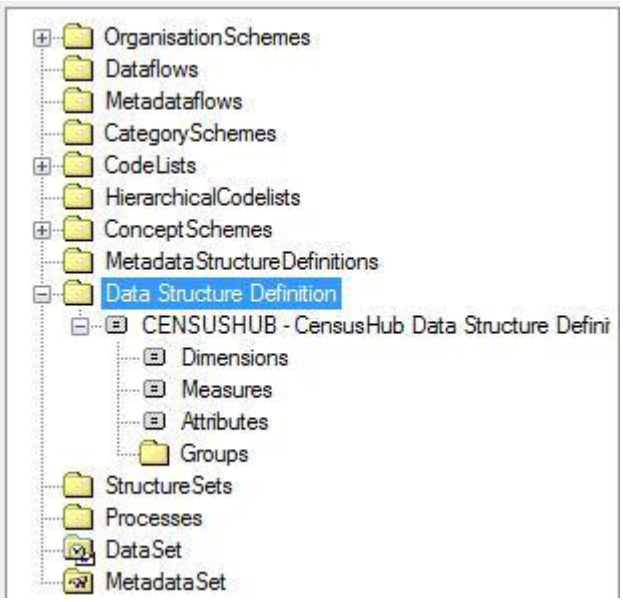
3.3.5 Navigation functions



These buttons allow navigating in the history of selected tree items. There are also navigation buttons on detailed view, associated to referenced artefacts. By using them, the referred artefact, if in memory, can be automatically selected in the tree.

3.4 Tree view

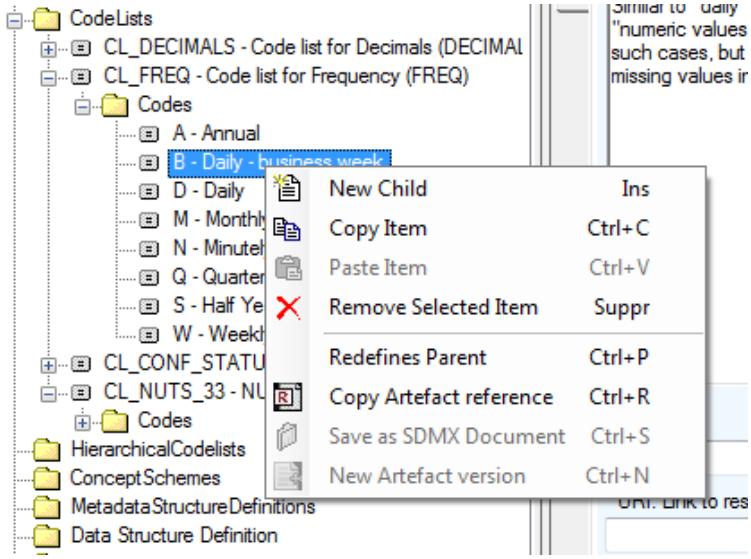
The tree view presents the collection of artefacts loaded in memory. These artefacts are ordered by types. Each artefact is identified by its id, name and version (if other than 1.0). Sometimes, items of artefacts are also presented in the tree.



3.4.1 Quick search

Searching in tree for a specific ID or a specific name is done with the quick search area. The search type is specified by a radio button. The text to search is specified in the associated text box and the search function is activated by pressing “Return” or with the search button action. The star “*” character can precede and/or follow the text to indicate wildcards.

3.4.2 Popup Menu



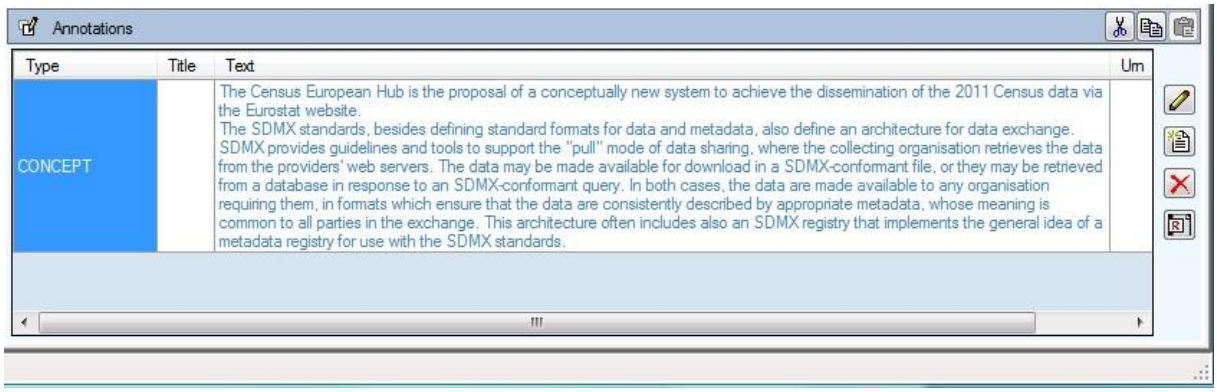
The tree popup menu operates on selected tree item. Menu choices are activated according to the type of the selected item.

The popup is activated by clicking the right button of the mouse inside the tree area.

Popup Menu choice	Operation
New Child	Creates a new Artefact item in selected artefact collection or creates a new specialized item inside the selected artefact.
Copy Item	Copy in memory the content of the selected item (code, concept, category or process step)
Paste item	The previously copied item is inserted after the selected item in tree.
Remove Selected Item	Removes selected item and all associated children. A confirmation is requested.
Redefine Parent	Redefines the parents hierarchy of the Artefact items. Examples are code list, categories and concepts.
Copy Artefact Reference	Same as in utility menu. This creates in the MS-Window clipboard the urn reference of the artefact or its sub item selected in the tree view.
Save as SDMX document	Save selected artefact as an SDMX document. If an artefact type container is selected, all artefacts held by the container are included in the created document.
New Artefact version	Creates a copy of the selected artefact. The copy has its version number updated.

3.5 Annotations View

The annotation view displays annotation(s) attached on the selected item of the tree view or of the detail view. Annotations can be managed through the annotation view buttons.


Annotations
Type
Title
Text
Urn
CONCEPT
The Census European Hub is the proposal of a conceptually new system to achieve the dissemination of the 2011 Census data via the Eurostat website.
The SDMX standards, besides defining standard formats for data and metadata, also define an architecture for data exchange. SDMX provides guidelines and tools to support the "pull" mode of data sharing, where the collecting organisation retrieves the data from the providers' web servers. The data may be made available for download in a SDMX-conformant file, or they may be retrieved from a database in response to an SDMX-conformant query. In both cases, the data are made available to any organisation requiring them, in formats which ensure that the data are consistently described by appropriate metadata, whose meaning is common to all parties in the exchange. This architecture often includes also an SDMX registry that implements the general idea of a metadata registry for use with the SDMX standards.

The size of the annotation area can be redefined by the horizontal split bar located above the annotation title.

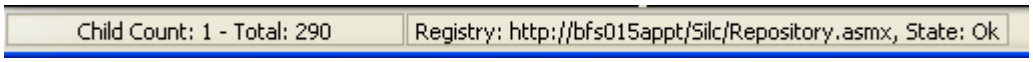
3.5.1 Functionalities

	Cut/Copy/Paste operations on Annotation. The operation is on the whole annotation, including the internationalized text.
	Edit selected Annotation in a specialized internationalised dialog box.
	Creates a new annotation attached to selected item.
	Deletes selected annotation.
	Creates an annotation of type “ARTEFACT_URN” using the last urn copied.

3.6 Status bar

The Status bar delivers information according to context.

3.6.1 Tree Childs count



According to selected item, the status bar displays the count of children items (sub level and total of sub-items.

3.6.2 Registry connection status

It displays also the current connection to TinyRegistry with its state.

3.7 Detail view

The detail view reflects the currently selected item in the tree. According to the complexity of the content it can be a tabbed view or a simple view.

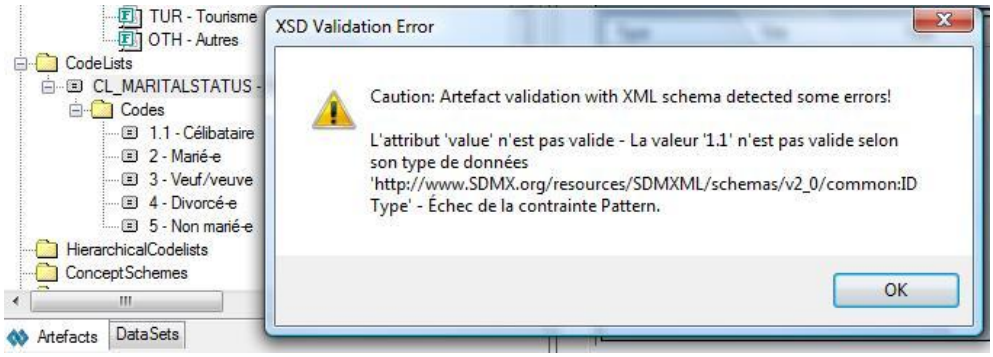
3.7.1 Read Only

If the artefact is signalled with the “Is Final” attribute, the edit operations are not allowed.

It is possible to uncheck the attribute in order to edit attributes. This is on the responsibility of the user.

When setting the “IsFinal” attribute the application performs an XML validation according to SDMX Structure schema. Detected errors are reported an a dialog box.

Note: to fully uses navigation possibilities the SDMX 2.0 schemas must be slightly modified to add some missing information like for example referred concept scheme version in DSD references.



4 Internationalization support

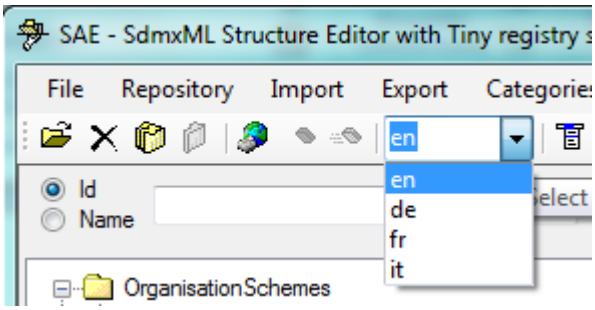
The application is designed to work with a set of defined languages for artefact texts.
The display of the language can be modified on the fly through the toolbar functions.

4.1 Language set definition

The set supported by the application is defined in the configuration file of the application. All languages codes of the supported set are defined separated by semi-column in the following configuration tag.

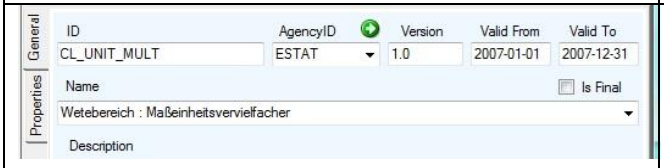
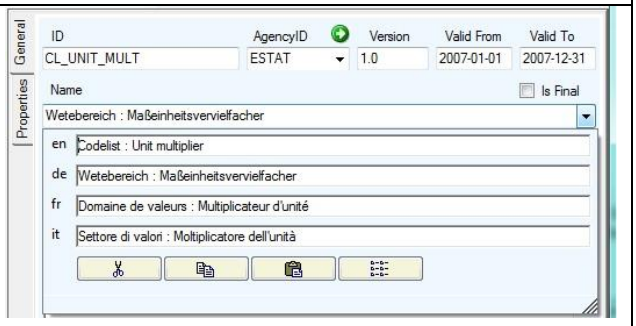
```
<add key="SupportedLanguages" value="en;de;fr;it"/>
```

The language abbreviation reflects the SDMX “xml:lang” attribute. The first language code of the set is used by default during application start. Language can be switched on the fly by selecting it in the toolbar language combo box.



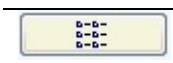
4.2 Multi Language edition

During text editing, the text in the current language can be typed directly in the text box, if other languages must be captured too, the text box offers a button on its right to expand edition to all supported languages.

Current language edition	Multi Language activated
	

Functionalities

Cut/Copy/Paste operations on Internationalized text. The operation is on the whole set of languages.



This function duplicates text currently written only for one language in the others text boxes. So the same text is assigned to all languages. Holds “Control Key” to force overwriting.

Convention

When all texts are same, the SDMX “xml:lang” attribute is not reflected when saving the artefact document. And during a read operation, a text without “xml:lang” attribute will be reflected in all language set.

This is the default mode of operation of the editor. It can be changed if in the configuration file of the editor the tag `<add key="TextLang" value= ---` is defined.

The possible values are:

Value	Signification
Strict	interprets texts without specified language as english, serialize according to defined languages
Expand	Text without a specified language are duplicated in all supported languages
Compress	If text is same for all languages it is written without language specification
Dual	Acts as if Expand + Compress are selected

5 Menu Repository

5.1 Introduction

The SAE Editor can work with the TinyRegistry repository service. This allows to navigate easily between artefacts and to manage them in a centralized manner.

The repository is a .NET 2.0 web service application to deploy under **IIS 5.1 or higher**.

To uses the repository, the SAE configuration file must contains the valid URL of the repository service page: `<add key="TinyRegistryUrl" value=""/>` If no URL value is specified the repository usage is ignored.

The repository manages structure artefacts by artefact type. It supports internationalization and must be configured according to SAE editor internationalisation subset support. Set of languages is specified when the repository is empty and cannot be changed once it is used.

A possibility exists for managing aggregated artefacts in a document. The Full iD is the one of the artefact selected when submitting the whole project.

For more details about the services please consult *TinyRegistry Repository service* documentation.

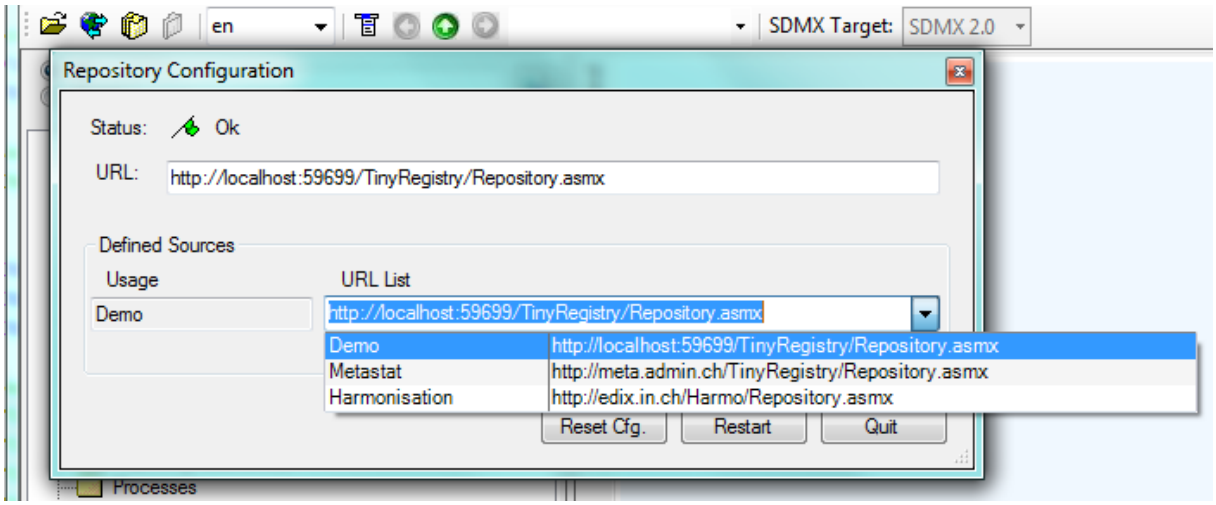
5.2 Usage from SAE Editor

5.2.1 Menu Repository – Define repository WS source…

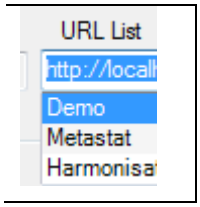
This menu activates a dialog box displaying the URL of the TinyRegistry service.

The current status is displayed as well as the URL. It is possible to restart or reset the service proxy, the status is updated accordingly.

A list of alternate repository can be defined in the application configuration file.



Reset Cfg.	Restart proxy access to Repository service by using the url specified in configuration file
Restart	Restart proxy access to Repository service by using the url specified by the user in the URL textbox.



Selects another repository URL. It is activated for the next accesses.

The optional list of alternate repository is given in configuration file with the tag `TinyRegistryAlternateUrl`

Each alternate repository is given as two items: usage – urn. All elements are separated by semi-column.

Sample:

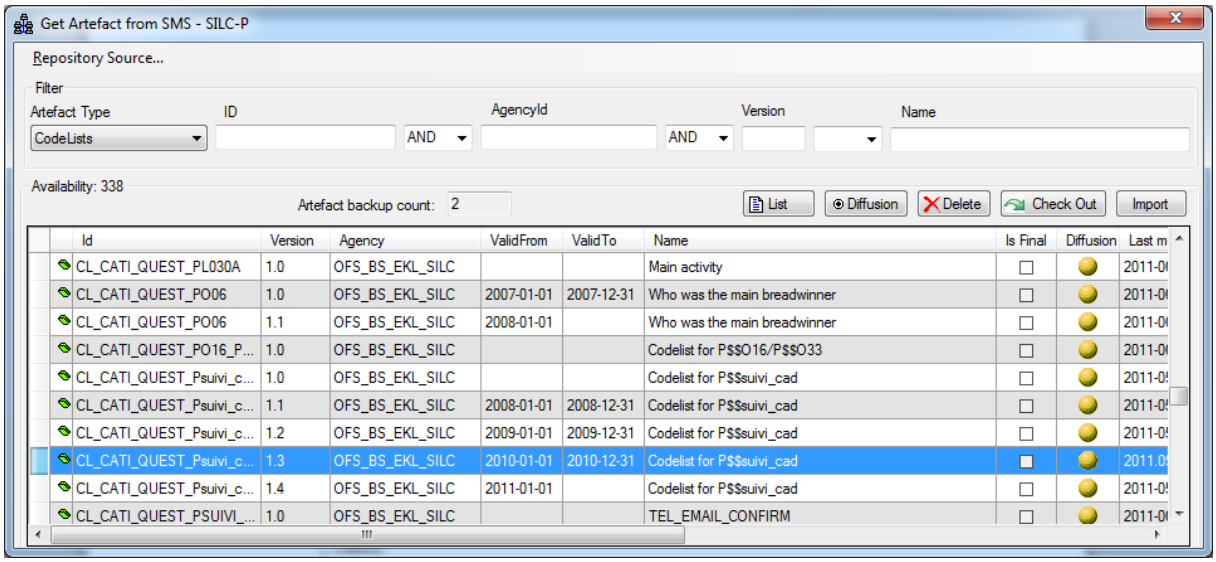
```
    <add key="TinyRegistryUrl"
value="Demo;http://localhost:59699/TinyRegistry/Repository.asmx" />
    <!-- Presented as: usage;http;usage;http  -->
    <add key="TinyRegistryAlternateUrl"
value="Metastat;http://meta.admin.ch/TinyRegistry/Repository.asmx;Harmonisa
tion;http://edix.in.ch/Harmo/Repository.asmx" />
```

Note: For a **multiplexor mode** (distributing registries and owners between servers), please contact us for more detail on installation processes.

5.2.2 Menu Repository – Get Artefact

Through this menu a Repository browser dialog box is open. It allows browsing the different artefacts available. A basic filter mechanism is provided for the full id and name column.

As the dialog box is non-modal it is also possible to operate on the main application screen at the same time (for example to consult detail of imported artefact).

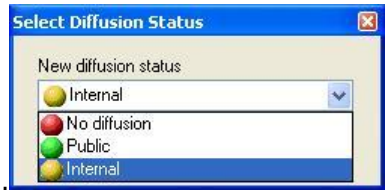

Get Artefact from SMS - SILC-P
Repository Source...
Filter
Artefact Type
ID
AgencyId
Version
Name
CodeLists
AND
AND
Availability: 338
Artefact backup count: 2
List
Diffusion
Delete
Check Out
Import
Id | Version | Agency | ValidFrom | ValidTo | Name | Is Final | Diffusion | Last m
CL_CATI_QUEST_PL030A | 1.0 | OFS_BS_EKL_SILC | | | Main activity
CL_CATI_QUEST_PO06 | 1.0 | OFS_BS_EKL_SILC | 2007-01-01 | 2007-12-31 | Who was the main breadwinner
CL_CATI_QUEST_PO06 | 1.1 | OFS_BS_EKL_SILC | 2008-01-01 | | Who was the main breadwinner
CL_CATI_QUEST_PO16_P... | 1.0 | OFS_BS_EKL_SILC | | | Codelist for P$$O16/P$$O33
CL_CATI_QUEST_Psuivi_c... | 1.0 | OFS_BS_EKL_SILC | | | Codelist for P$$suivi_cad
CL_CATI_QUEST_Psuivi_c... | 1.1 | OFS_BS_EKL_SILC | 2008-01-01 | 2008-12-31 | Codelist for P$$suivi_cad
CL_CATI_QUEST_Psuivi_c... | 1.2 | OFS_BS_EKL_SILC | 2009-01-01 | 2009-12-31 | Codelist for P$$suivi_cad
CL_CATI_QUEST_Psuivi_c... | 1.3 | OFS_BS_EKL_SILC | 2010-01-01 | 2010-12-31 | Codelist for P$$suivi_cad
CL_CATI_QUEST_Psuivi_c... | 1.4 | OFS_BS_EKL_SILC | 2011-01-01 | | Codelist for P$$suivi_cad
CL_CATI_QUEST_PSUIVI_... | 1.0 | OFS_BS_EKL_SILC | | | TEL_EMAIL_CONFIRM

The grid content reflects the different information's managed by the repository. The type of artefact to list can be selected through the Artefact type combo box.

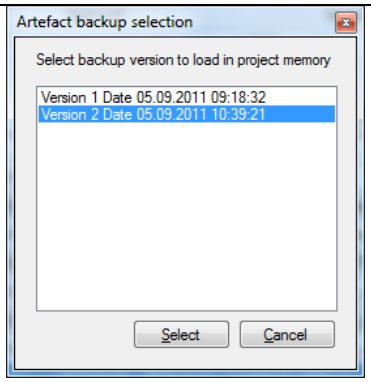
If some backup of selected artefact exists it is reflected in the Artefact backup count information box.

It is also possible to redefine repository source from this dialog. Dialog Menu: Repository Source…

Grid functions

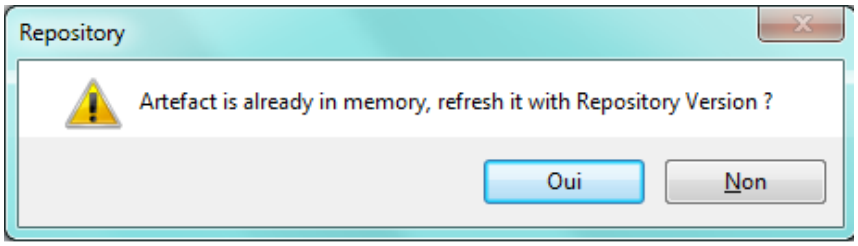
List	Activate the list request, used at first demand and to obtain a filtered list. Note: when the type of artefact is changed the list action is automatically called.
Diffusion	Redefines diffusion status. Default is Internal 
Import	Loads in the application the artefact selected in the grid. A double mouse click on a grid item activates the same functionality.
Check Out	Loads in the application the artefact selected in the grid. The artefact is locked in the repository and only the actual user will be allowed to submit an update of this artefact to the repository. This operation is password protected.
Delete	Remove selected artefact from Repository. CAUTION: uses appropriately..

Backup retrieval

When submitting an artefact to the registry it is possible to preserve the previous content already present in the registry as a backup. So, if some backup of selected artefact is available it is reflected in the Artefact backup count information box. Clicking on the backup count raises a dialog box that allows the user to select the backup to load instead of the current content of the selected artefact.	

5.2.3 Menu Repository – Reserve Artefact (check out)

It is also possible to reserve an artefact of the repository through this menu if it is already present in memory. This avoids using the repository list dialog. The system asks if replacement of artefact is needed.

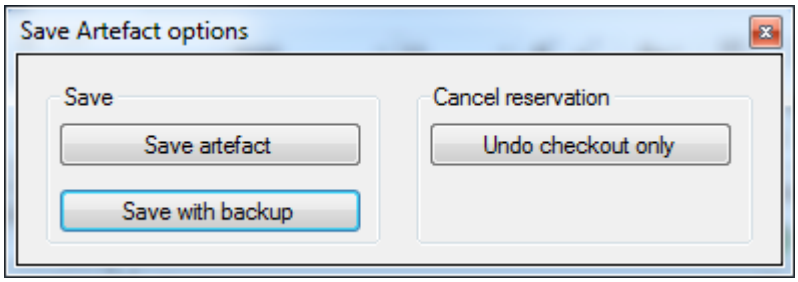


5.2.4 Menu Repository – Submit Artefact (check in)

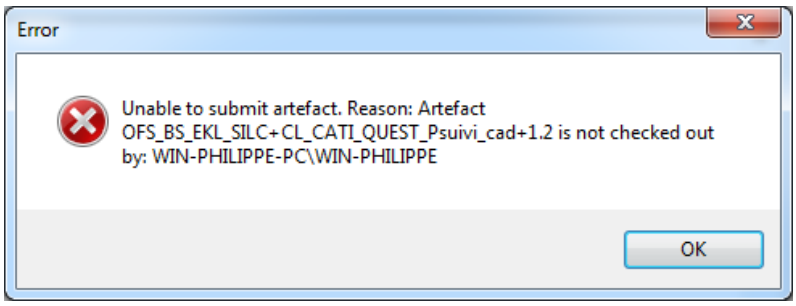
In order to submit an artefact to the repository it is required, if the “ownership management” is activated in TinyRegistry configuration, to be member of the same agency group that the artefact agency or to be an administrator.

The submitted artefact is the one currently selected in the main tree view. If the artefact is not yet in the repository it is automatically submitted. If the artefact already exists in repository it must have been previously reserved (check out) by the same user. Once successfully submitted, the reservation is cleared.

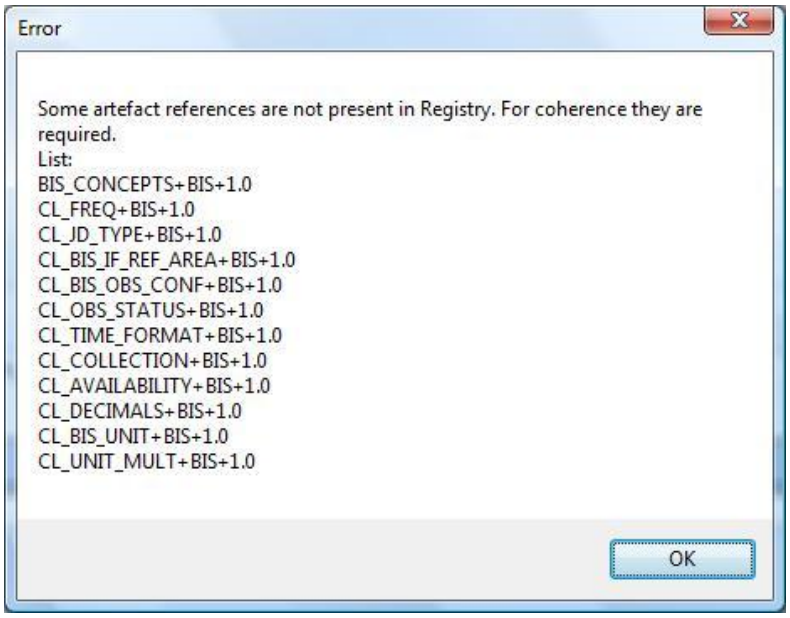
A dialog box is raised if the artefact is already present in registry:



If the artefact was not previously reserved an error message is displayed:



If the artefact refers others artefacts they must already be in registry in order to store it.



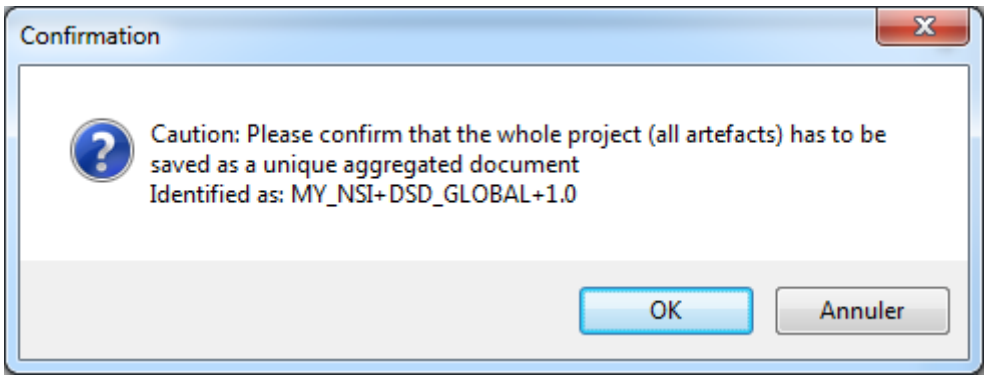
Note: It is possible to cancel a previous reservation (a check-out) without changing the artefact located in repository. Simply answers “Undo checkout only” in the proposed dialog box.

5.2.5 Menu Repository – Submit aggregated Artefacts to Repository

When “ownership management” is activated in TinyRegistry configuration only administrators can use this functionality. It allows saving as a unique document, the whole memory project. If some of the artefacts of the project are already present in repository as single document,

The artefact full ID that is used to identify the document in the repository is the one of the artefact currently selected in the tree view.

As it is not a standard operation, a confirmation is required:


Confirmation
Caution: Please confirm that the whole project (all artefacts) has to be saved as a unique aggregated document
Identified as: MY_NSI+DSD_GLOBAL+1.0
OK
Annuler

5.2.6 Menu Repository – Update Texts of Finalized Artefact

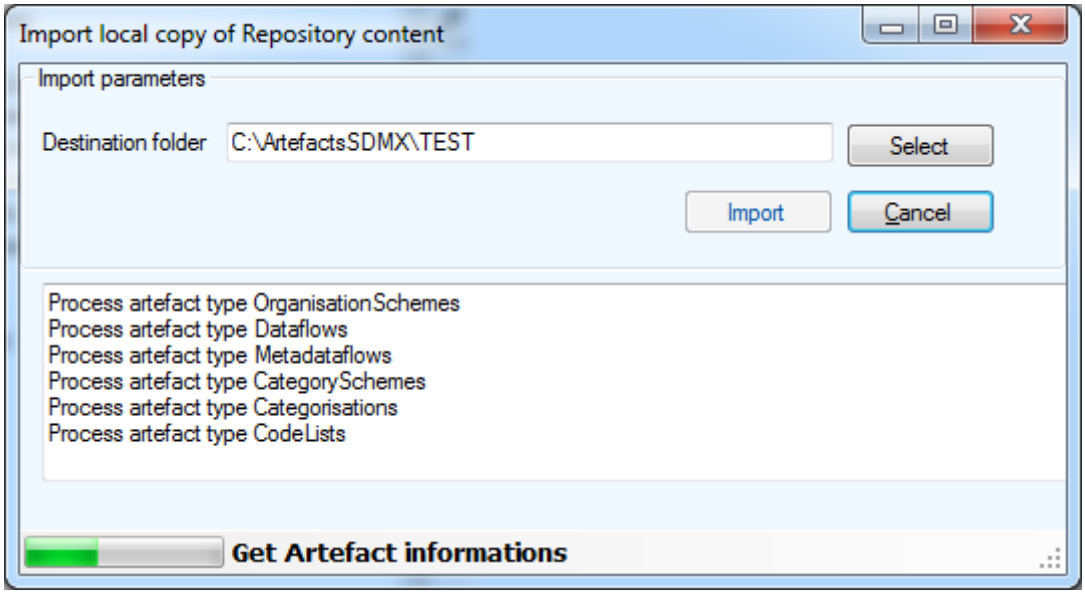
This command allows to update texts of an already finalized artefact located in registry.

Name and Description are updated for the artefact. And for code list, category scheme and concepts schemes, name and description of embedded items are also updated.

The artefact with updated text must be selected in tree before activating the functionality. The user has to be be the owner of the artefact.

5.2.7 Menu Repository – Import local Image copy

This command activates a dialog box where the user specifies the directory where the repository content has to be copied. The specified directory must be empty. The user can then activate the copy process. The artefacts are copied and grouped by type in sub directories of the specified target.



The process can be cancelled with the cancel button.

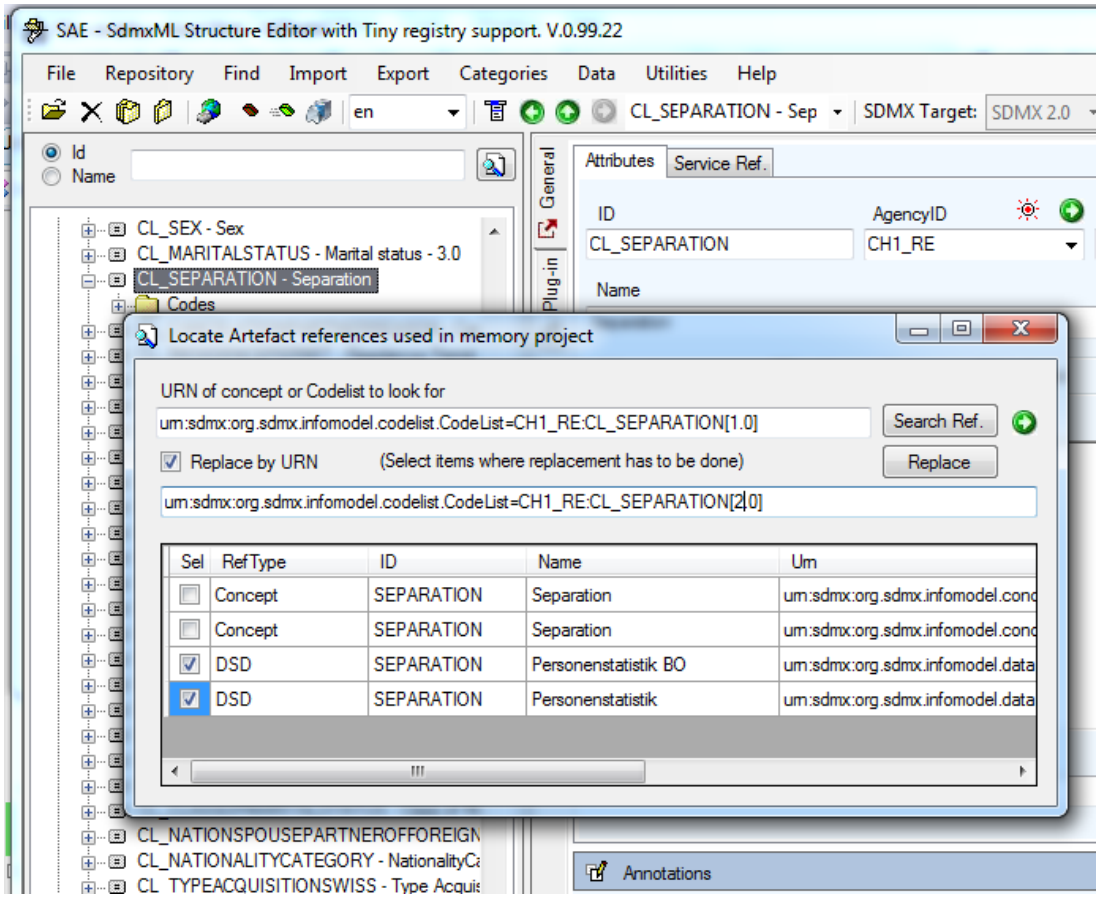
Note: In multiplexed mode the import process differs and is done by Owner group refered metadata,

6 Other Menus

6.1 Menu Find

6.1.1 Locate References...

This function locates the references of the code list or concept selected in main tree or given manually in the non-modal dialog box URN field.



A non-modal dialog box is displayed with all references found in project memory (in ConceptScheme, HierarchicalCodelist, DSD or MSD). It is then possible to activate automatically the detailed view of the artefact item holding the selected reference.

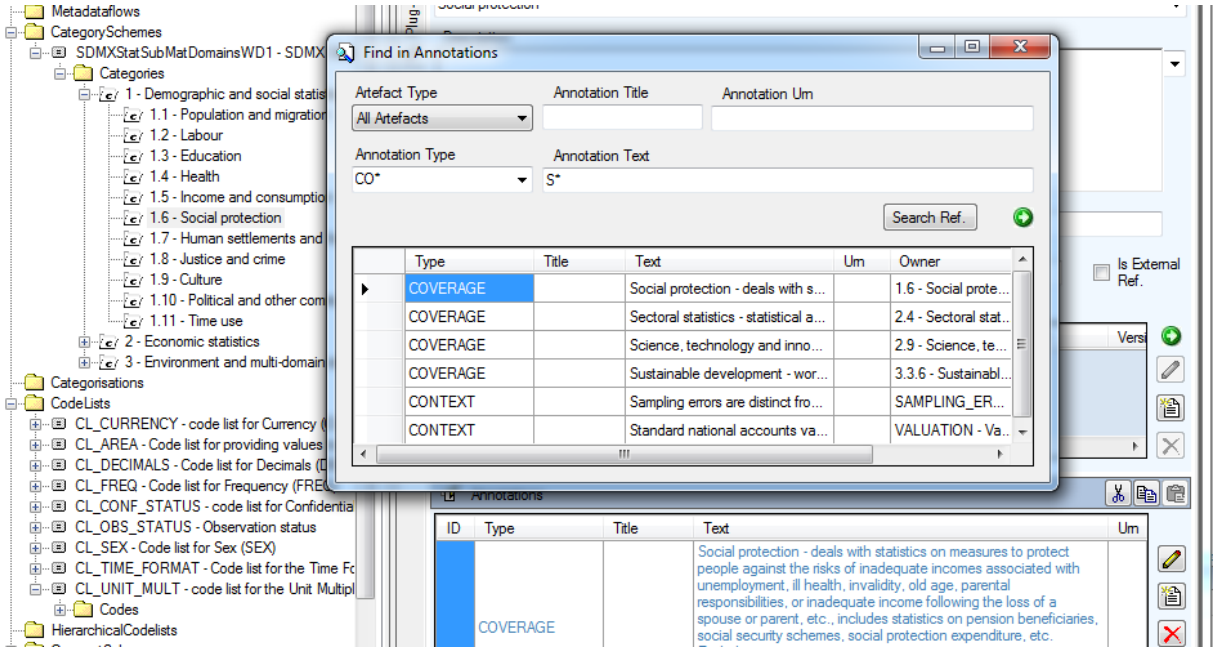
Replacement of reference in selected items shown in displayed list is also possible.

Search Ref.	This button activates a search of reference usage for concept or codelist specified in the urn textbox. NOTE: this is done automatically if the codelist or concept is selected in tree view before activating the menu
	Activates detail view of the artefact item holding the reference. The dialog box remains open, so multiple references can be browsed easily.
Replace	If the check box “replace by URN” is it is possible to select in list the references to modify by the new specified URN. To obtain the urn of an item as codelist, concept or conceptScheme, simply selects item in main tree and activates popup menu “Copy artefact reference” then uses “CTRL-V to paste item in URN textbox,

To look for another reference, simply change the selection of concept or code list in tree view and reactivate the menu. The dialog box is refreshed with the new material.

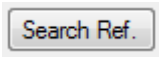
6.1.2 Find Annotations...

This function locates the references of the annotations present in memory. Annotations are selected according to specified criteria. They are listed in the modeless dialog grid.



Search criteria	Default value	Comment
Artefact type	ALL	Specifies artefact type by selecting it in combo box
Annotation title	unspecified	Specifies title to look for. Wildchars (*) allowed at start and end of specification
Annotation URN	unspecified	Specifies urn in annotation to look for. Wildchars (*) allowed at start and end of specification
Annotation Type	unspecified	Specifies type of annotation to look for. Wildchars (*) allowed at start and end of specification
Annotation Text	unspecified	Specifies text in annotation to look for. The text is searched for currently selected language, Wildchars (*) allowed at start and end of specification

The non-modal dialog box grid is then filled with all references found in project memory. It is then possible to activate automatically the detailed view of the artefact item holding the selected reference to annotation.

	This button activates a search of annotations matching the criteria.
	Activates detail view of the artefact item holding the annotation. The dialog box remains open, so multiple references can be browsed easily.

To look for another annotation reference, simply change the search criteria and reactivate the search reference function. The dialog box is refreshed with the new material.

6.2 Menu Import / Export

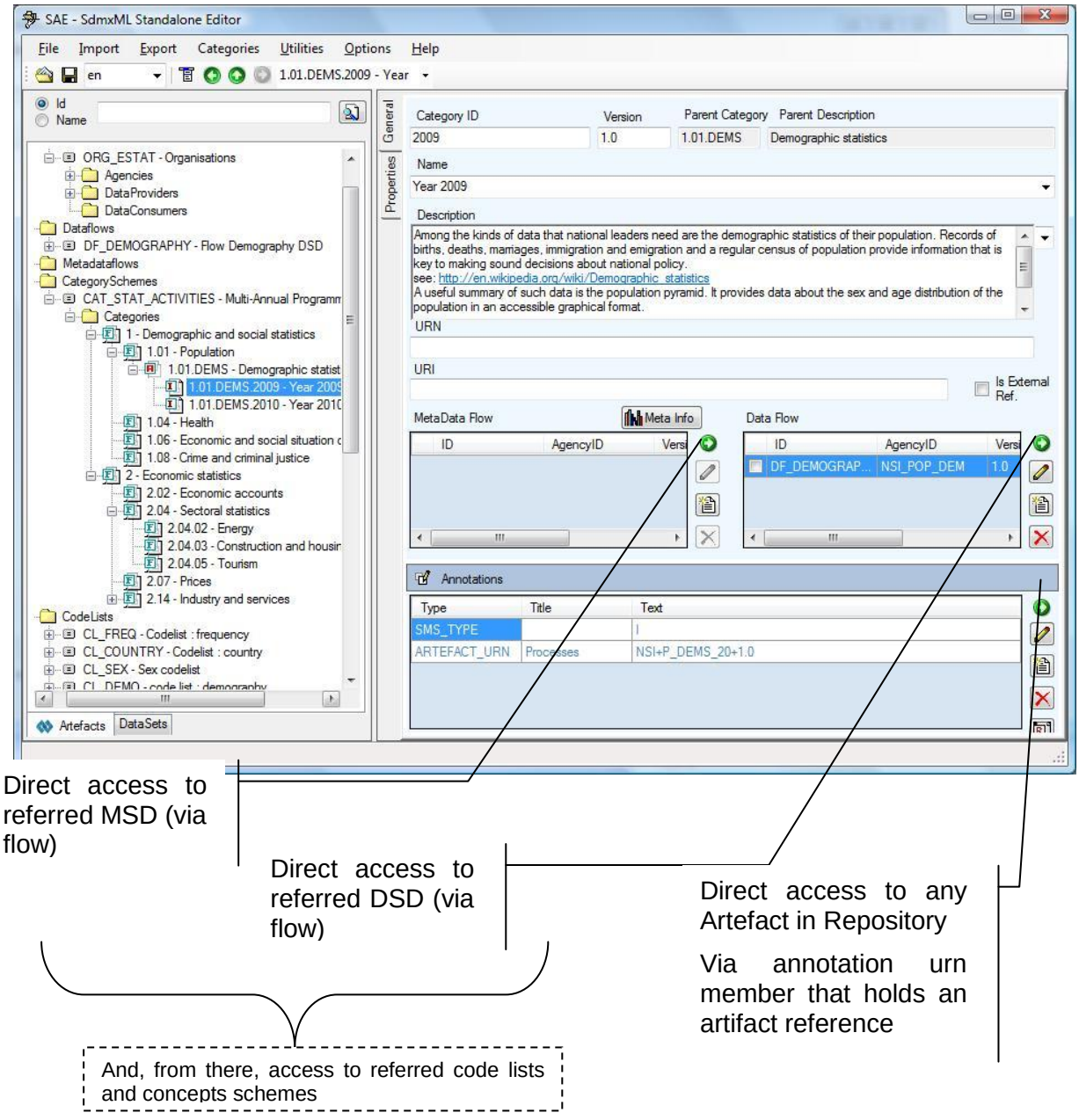
See Excel paragraph.

6.3 Menu Categories

It is nice to have coherent navigation entry points. For this, the Categories menu allows you to define categories artefact to load from repository in order to navigate between artefacts through different views.

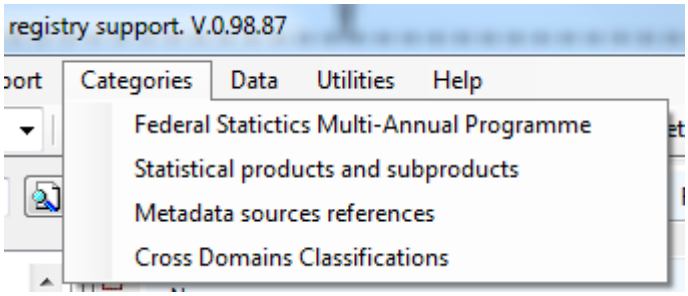
You can provide in the application configuration file some full ID to load specific categories designed for navigation like: category for statistical activities, category for transversal codelists …

Then, by clicking on navigation arrows of the detail view you can directly load and access referred artefacts.


SAE - SdmxML Standalone Editor
File Import Export Categories Utilities Options Help
Direct access to referred MSD (via flow)
Direct access to referred DSD (via flow)
Direct access to any Artefact in Repository
Via annotation urn member that holds an artifact reference
And, from there, access to referred code lists and concepts schemes

6.3.1 Sub menus definition:

This functionality allows creating entries to different category schemes that represents view point to the metadata material held in registry. See menu sample below:

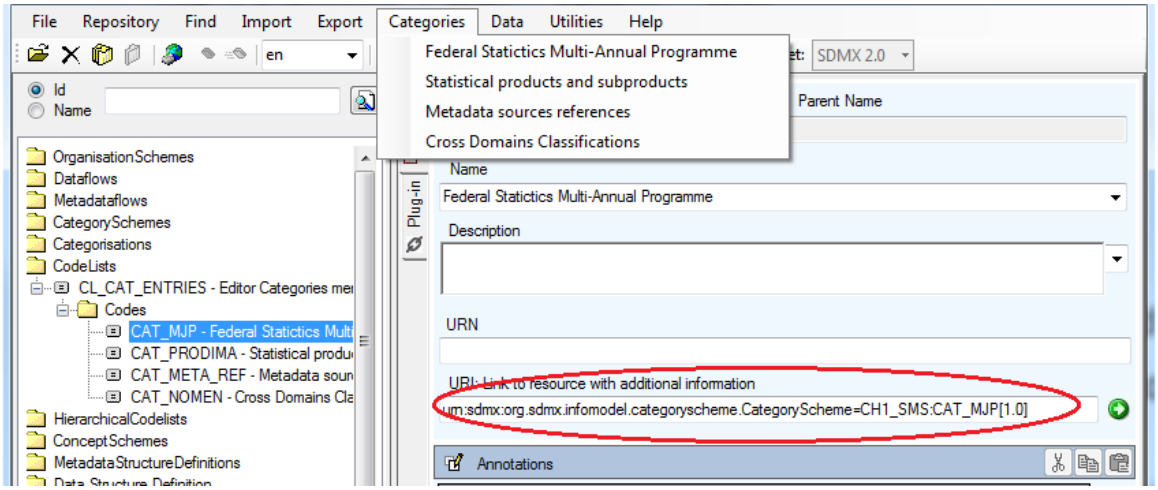


The sub-menu items must be defined in a SDMX codelist located in the repository.

The code list reference is given in the application configuration file under the tag: "`CategoryRootFullId`". Example:

```
<add key="CategoryRootFullId" value="CH1_SMS+CL_CAT_ENTRIES+1.0" />
```

Each code of the code list represents a menu item entry. The name of the code is used as text of the menu item. The URI field must contains an SDMX reference to the artifact to load from the repository when this item is activated.



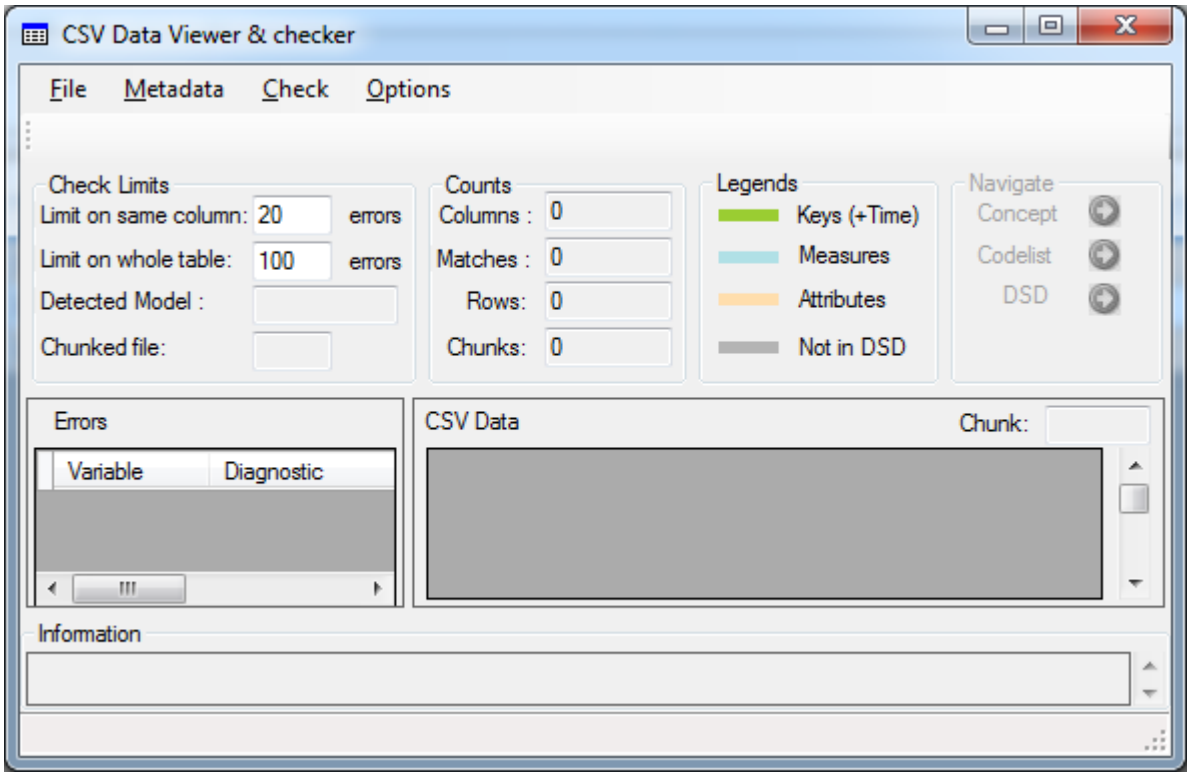
6.4 Menu Data

6.4.1 SQL...

(Experimental)

6.4.2 CSV…

This function raises a non-modal dialog for binding CSV data to corresponding metadata.



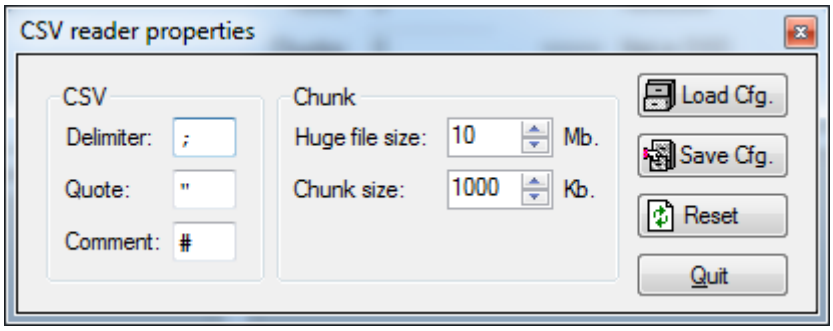
Dialog menu

Menu	Menu Item	Description
File	Load	Raise a dialog box to select CSV file to load on CSV grid. Note: csv reader is configurable (see option menu)
Metadata	Bind	Bind CSV columns to corresponding DSD selected in main tree. The column order is meaningless. The binding is done according to CSV header matching concept ID.
Check	Full	Once correctly bind to metadata it is possible to check columns content according to codelist items.
Check	Current chunk	For huge files, it checks only the chunk part of the file that is currently in memory,
Check	Load previous diagnostic	Reload a diagnostic of checking previously done.

Options	CSV format	Raise a dialog box used to specify CSV parameters as well as huge file detection limit.

It is currently required that the first row of the CSV file contains header with column name matching concept referred in DSD.

CSV Format menu

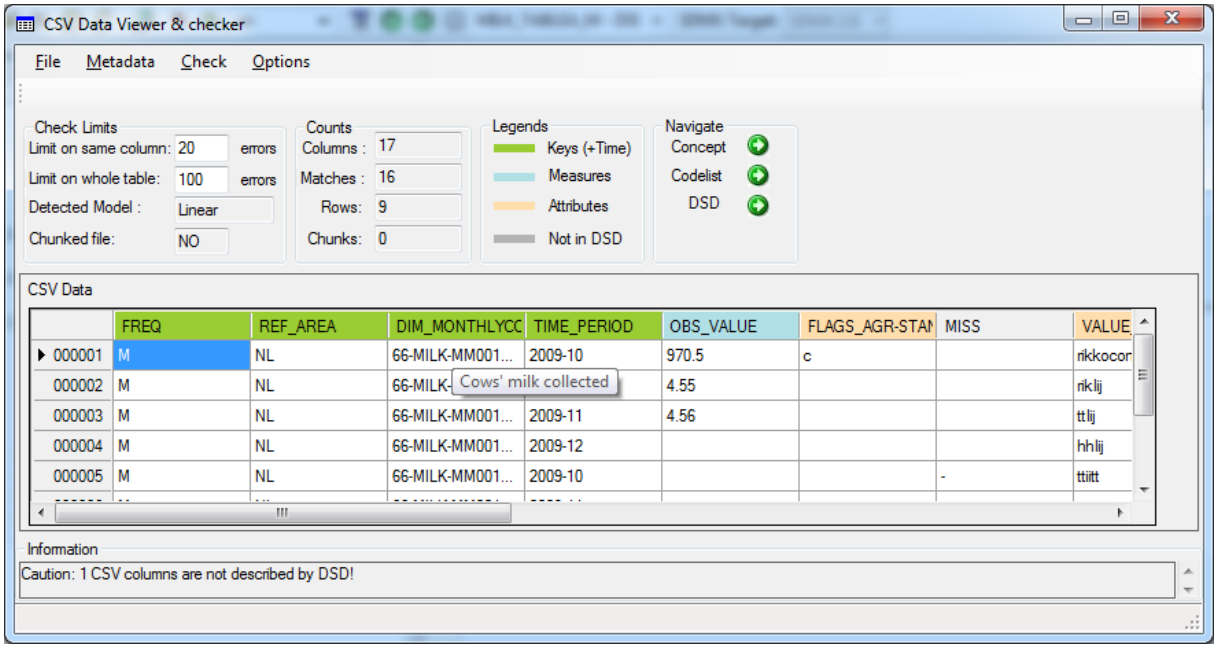


Huge file size defines the maximum size of a CSV file that can be held in application memory. For bigger files, the application cut the source file as a set of chunk. It is then possible to navigate between the chunks.

Data Sample

Load DSD information “MILK_TABLEA_M+ESTAT+2.1.xml” contained in SampleFiles, then activates Menu Data – CSV and load csv file “MILK_TABLEA_M_template_sdmx_data.csv”contained in SampleFiles\data.

Then activates dialog menu Metadata-Bind. The grid reflects binding as follow:



When the cursor is placed in a cell it displays the content name for cell containing code and the header cell concept name for header cell.

From the selected cell, it is possible to directly navigate in editor detail view of associated item for concept, code list or DSD item.

6.5 Menu Utilities

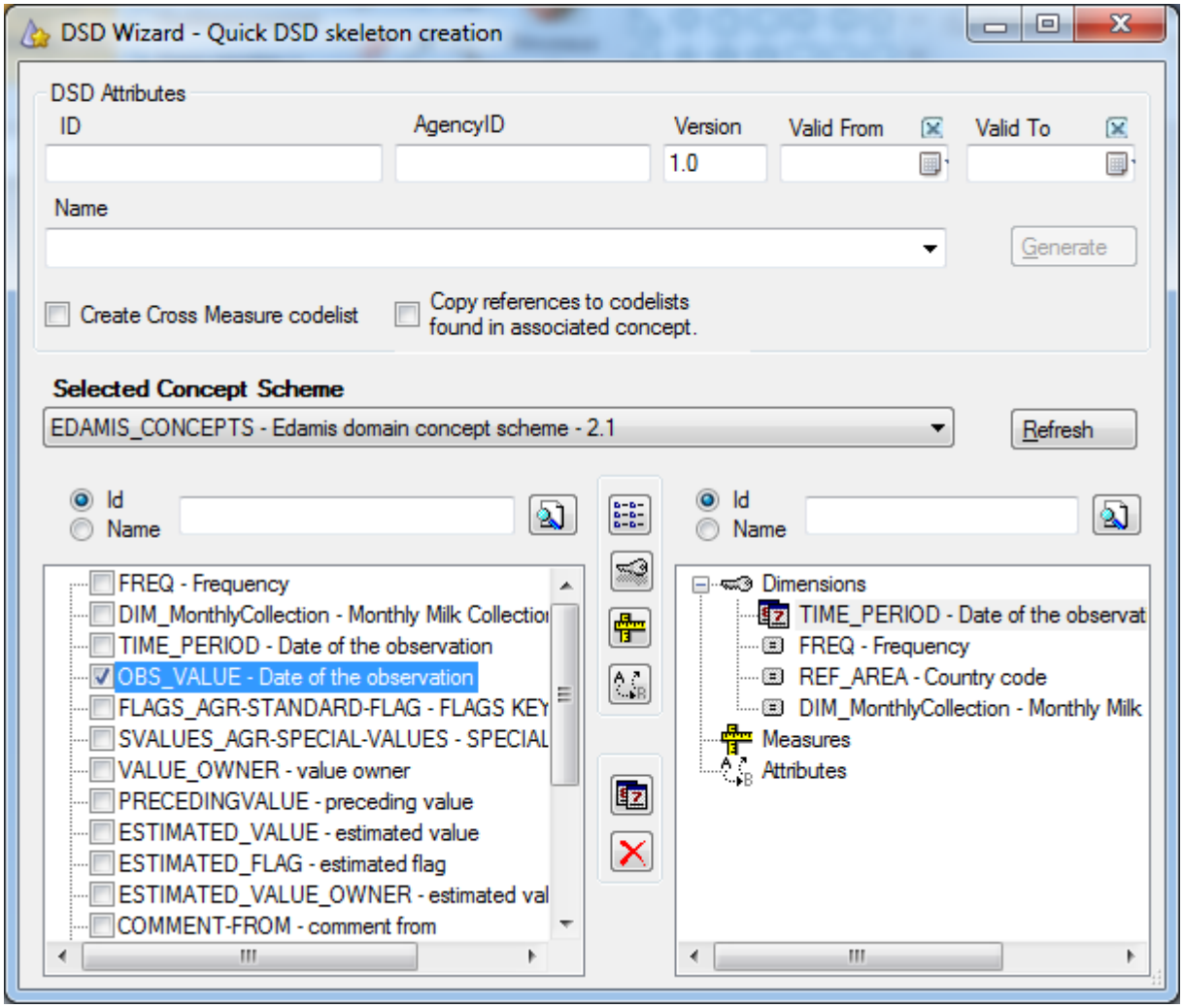
6.5.1 Copy Artefact Ref.

This function creates in the MS-Window clipboard the urn reference of the artefact or its sub item selected in the tree view. The urn can be copied as text or as annotation of type ARTEFACT_URN.

Sample of urn: *urn:sdmx:org.sdmx.infomodel.codelist.Code=ESTAT:CL_GEO[1.1].IT*

6.5.2 DSD Wizard...

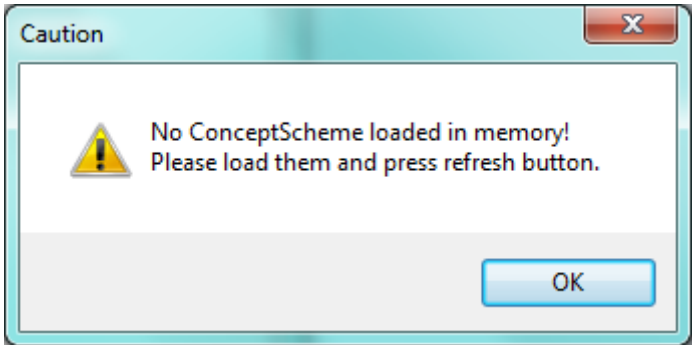
This function allows quick creation of a DSD skeleton. Optionally it creates also associated code list for Cross sectional measures (if any).
A modeless box is displayed in order to select concepts to associate to the DSD



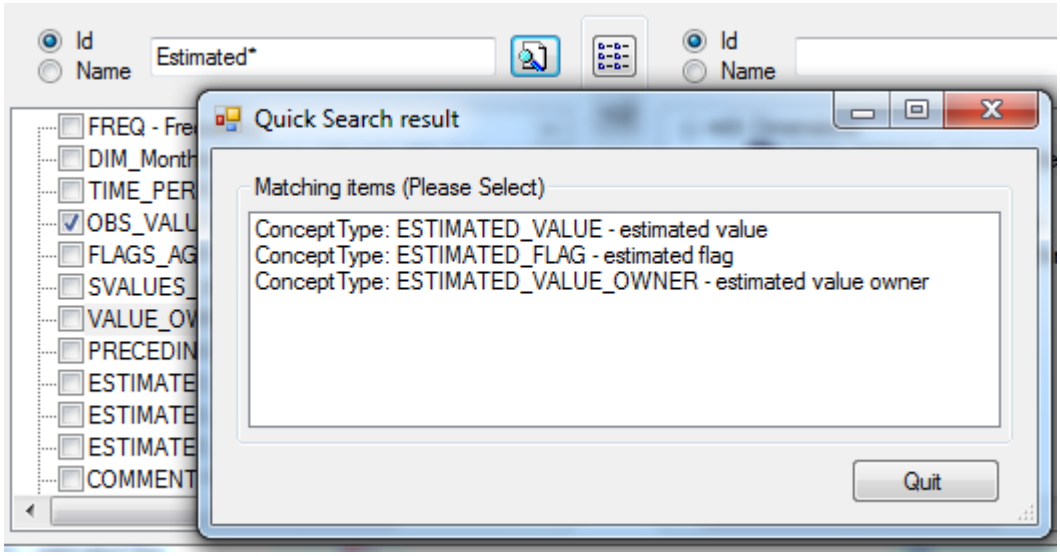
Proceed as follow:

1. In scheme combo box, selects conceptScheme holding concepts to use
2. Check concepts to distribute in a DSD domain (Dimensions, measures or attributes)
3. Transfers them by activating corresponding domain button
4. Proceeds for all domains and all schemes required.
5. In DSD Tree, reorders concepts with dragging function possibility
6. Optionally defines Time Dimension concept (select it in dimensions and press “time dimension” button)
7. Fills DSD attributes (ID, Agency ID, etc.)
8. Press Generate button. The DSD skeleton artefact is created and can be refined in main editor.

The concept schemes to use should be loaded in memory before using wizard. If none available an error message is displayed:



A basic search function on concept ID or concept name is available for each tree view:

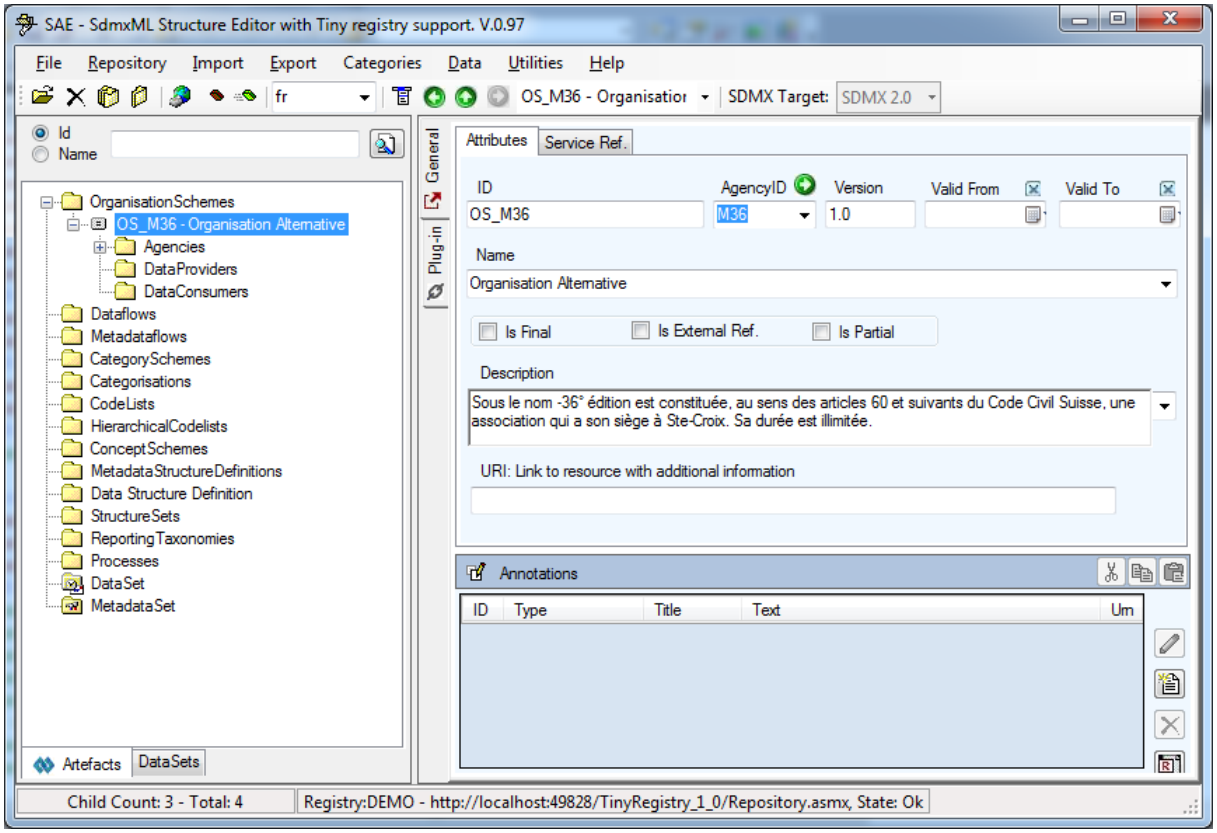


7 Artefacts detail view

7.1 Organization Scheme

7.1.1 Artefact detail

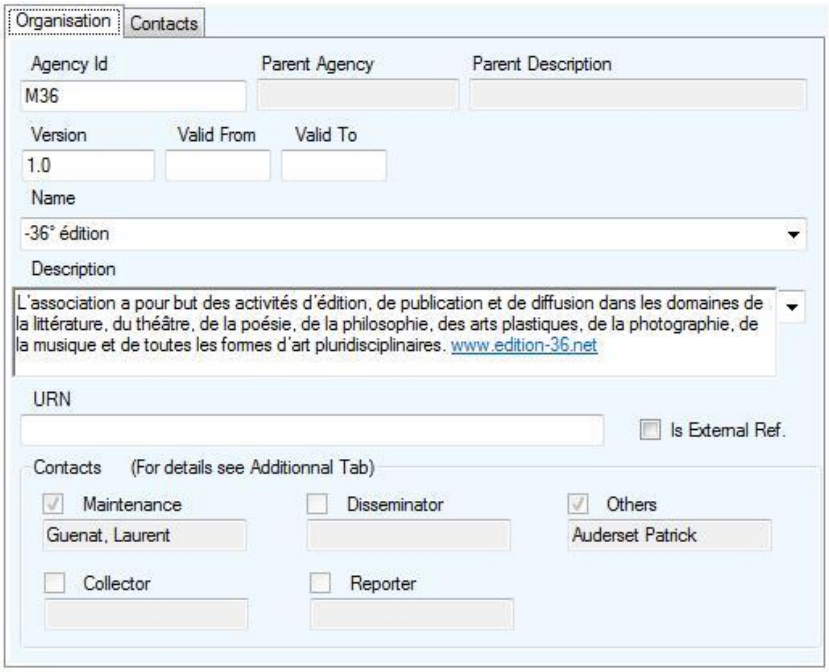
The detail screen manages the standard artefact attributes. Name and Description support multi language edition. The Service Ref. Tab is provided for future SDMX 2.1



7.1.2 Agency, Data Provider and Data Consumer detail

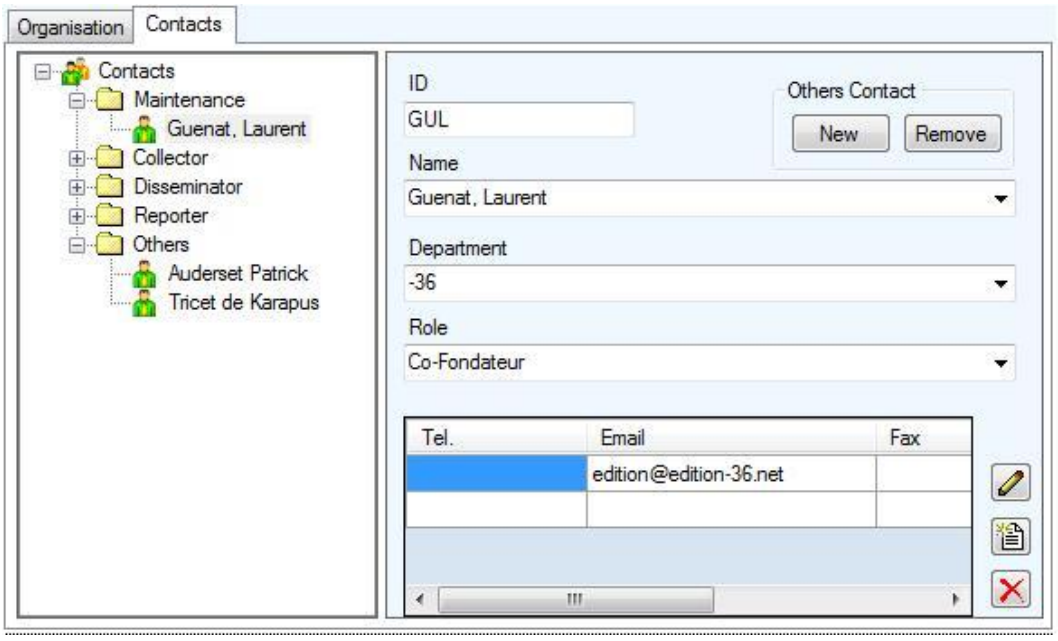
The tree reflects the defined hierarchy in case of hierarchic Agencies / data Providers or Data Consumers. The detail of agency or provider is developed in two tabs as follow.

7.1.3 Organisation tab

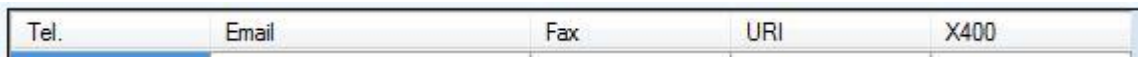


The tab allows specifying the element attributes. In case of hierarchy, the parent information is given as read only information. If any link is given in textual information, the link can be activated. A summary of contacts is also displayed in this tab.

7.1.4 Contacts Detail tab



Contacts are displayed in a tree. The detail of a contact reflects the selected item in the contact tree. As “Others contact” can be a collection of contact, A group box allows to create or remove others contacts.



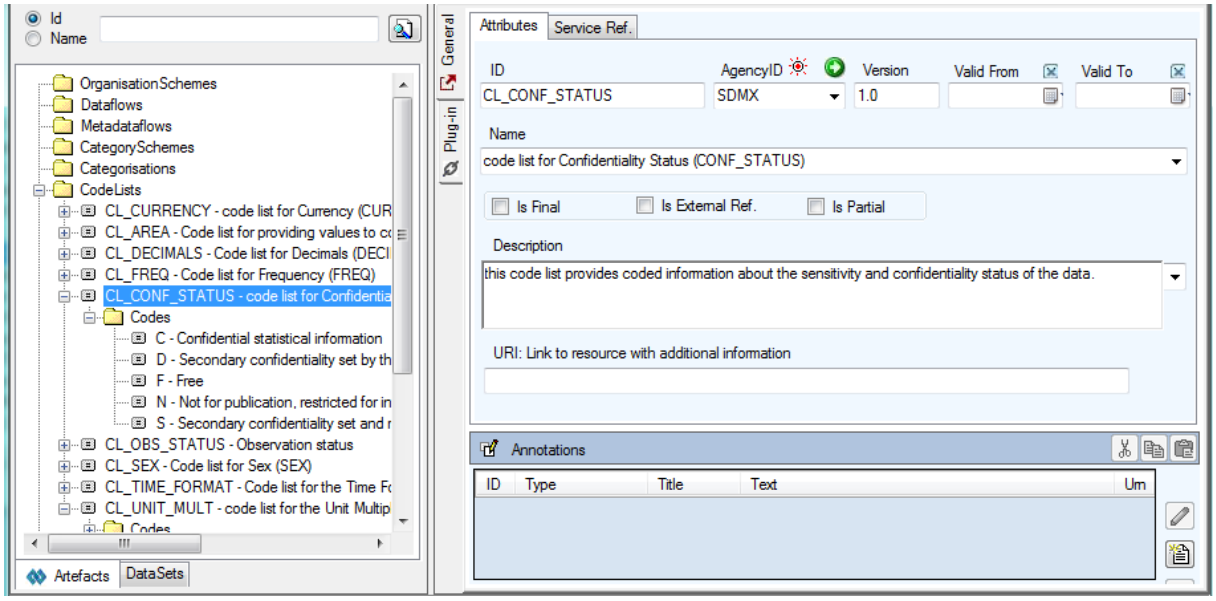
A grid groups these attributes. They can be created as follow.

	Edit selected row of the group information. A specialized dialog box is open in order to specify the group row attributes.
	Creates a new row for information group.
	Deletes selected row in group.

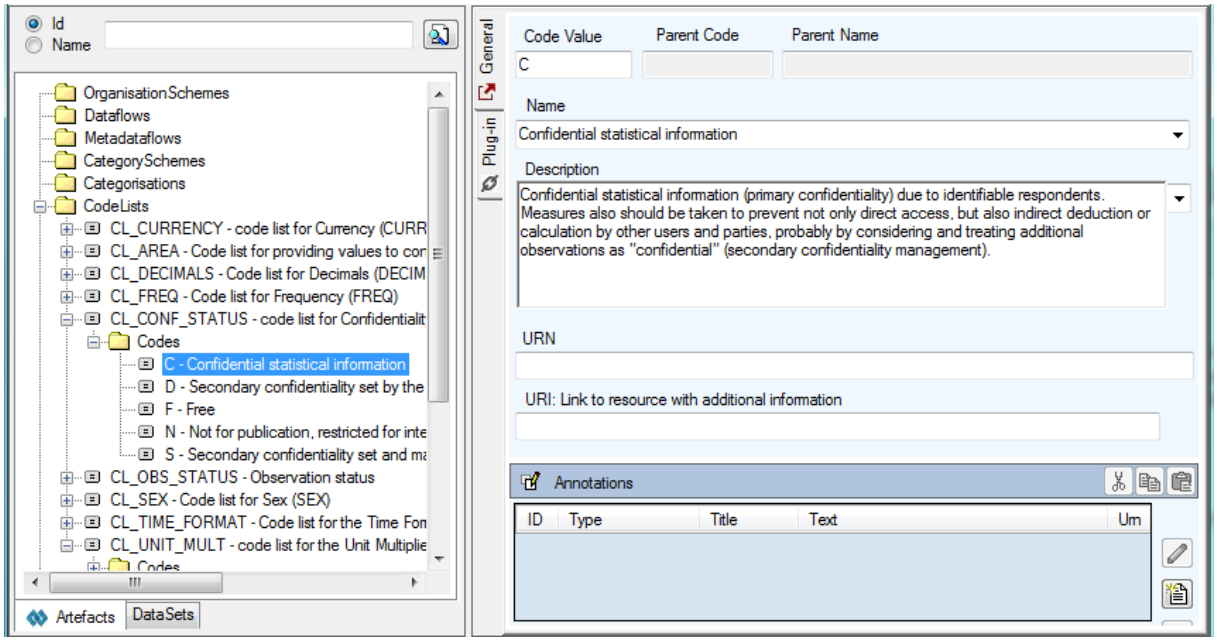
7.2 Code Lists

7.2.1 Artefact detail

The detail screen manages the standard artefact attributes. Name and Description support multi language edition.
The URI box allows a link to additional information. When filled, a navigation button is available. The checkbox "*Is Partial*" has no meaning in SDMX 2.0



7.2.2 Code detail

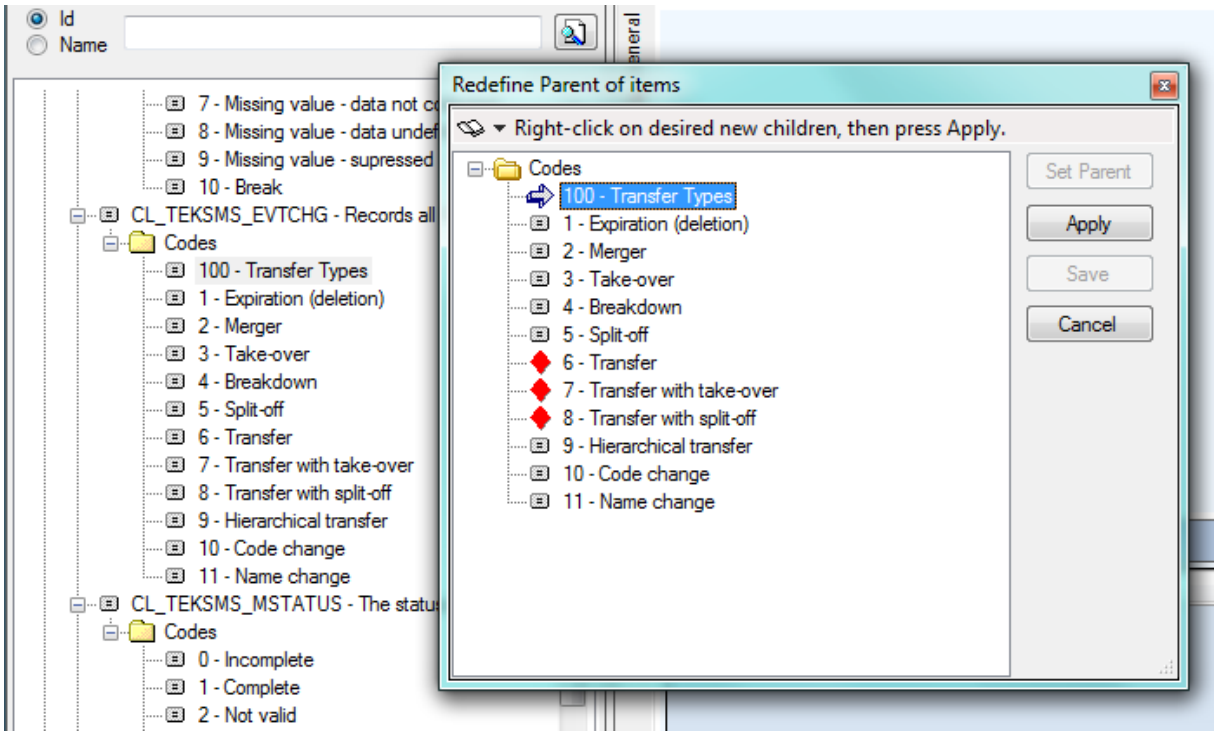

Id
Name
OrganisationSchemes
Dataflows
Metadataflows
CategorySchemes
Categorisations
CodeLists
CL_CURRENCY - code list for Currency (CURR
CL_AREA - Code list for providing values to cor
CL_DECIMALS - Code list for Decimals (DECIM
CL_FREQ - Code list for Frequency (FREQ)
CL_CONF_STATUS - code list for Confidentialit
Codes
C - Confidential statistical information
D - Secondary confidentiality set by the
F - Free
N - Not for publication, restricted for inte
S - Secondary confidentiality set and ma
CL_OBS_STATUS - Observation status
CL_SEX - Code list for Sex (SEX)
CL_TIME_FORMAT - Code list for the Time For
CL_UNIT_MULT - code list for the Unit Multiplie
Codes
Artefacts
DataSets
General
Plug-in
Code Value
Parent Code
Parent Name
C
Name
Confidential statistical information
Description
Confidential statistical information (primary confidentiality) due to identifiable respondents. Measures also should be taken to prevent not only direct access, but also indirect deduction or calculation by other users and parties, probably by considering and treating additional observations as "confidential" (secondary confidentiality management).
URN
URI: Link to resource with additional information
Annotations
ID
Type
Title
Text
Um

The detail screen manages the code attributes. The code hierarchy (if any) is presented in the tree view part. Parent information is also visible in the detail screen. Code name supports multi language edition.

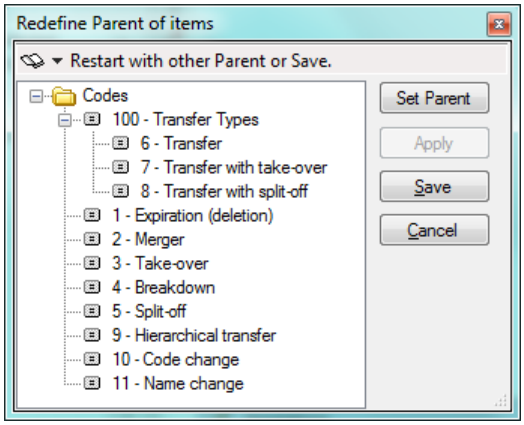
A Description box is provided, in SDMX 2.0 it is persisted in the document as an annotation of type "*NOTE*". Others annotation(s) attached to selected code are visible on the annotation panel.

7.2.3 Code hierarchy management

- The hierarchy of code is managed with the tree popup menu. Code can be added as child of selected item and can be removed with all associated children.
- Parent hierarchy can also be redefined with popup menu functionality.



1. Activate the dialog box via the popup menu
2. Select target parent in tree view and press button "*Set Parent*"
3. Select items to place under the selected parent with right click in corresponding tree items.
4. Press button "Apply" to see the result,
5. Redo the operations for another parent or save the result with the "*save*" button. Usage of the "*Cancel*" button discards all previous operations and closes also the dialog box.

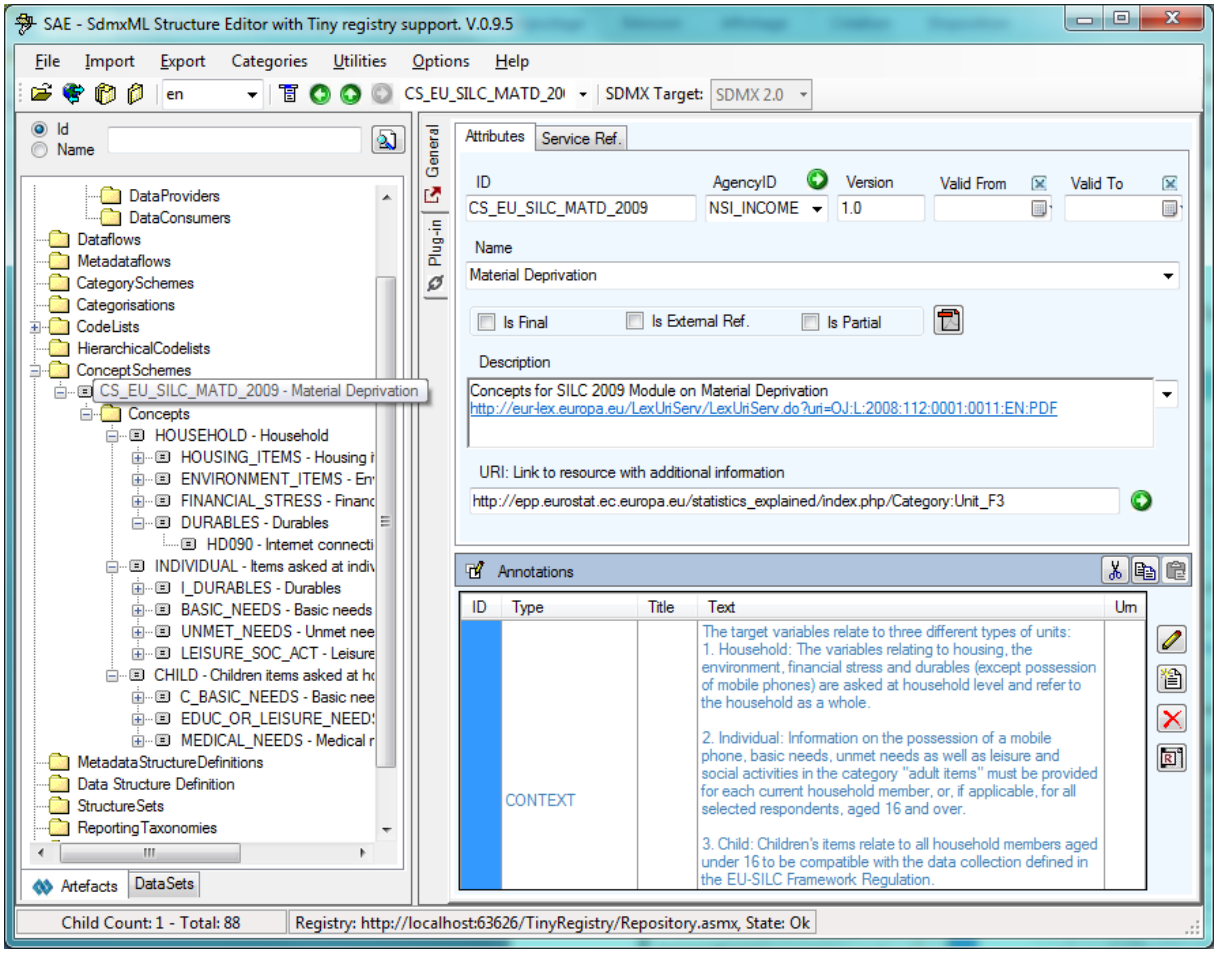


- Codes can be reordered (in current level) with drag and drop operation in tree. This is valid for the main tree view and for the "*redefine parent*" dialog box tree.

7.3 Concept Scheme

7.3.1 Artefact detail

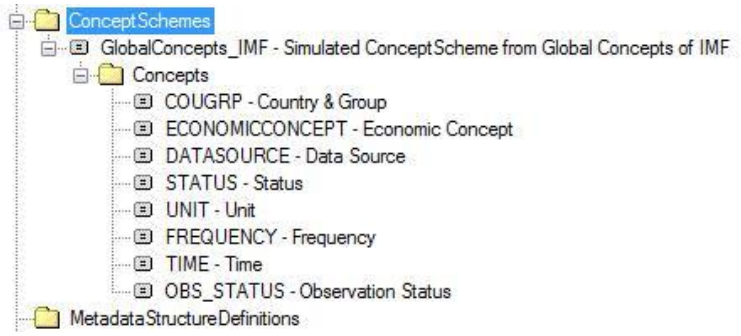
The detail screen manages the standard artefact attributes. Text and Description support multi language edition.



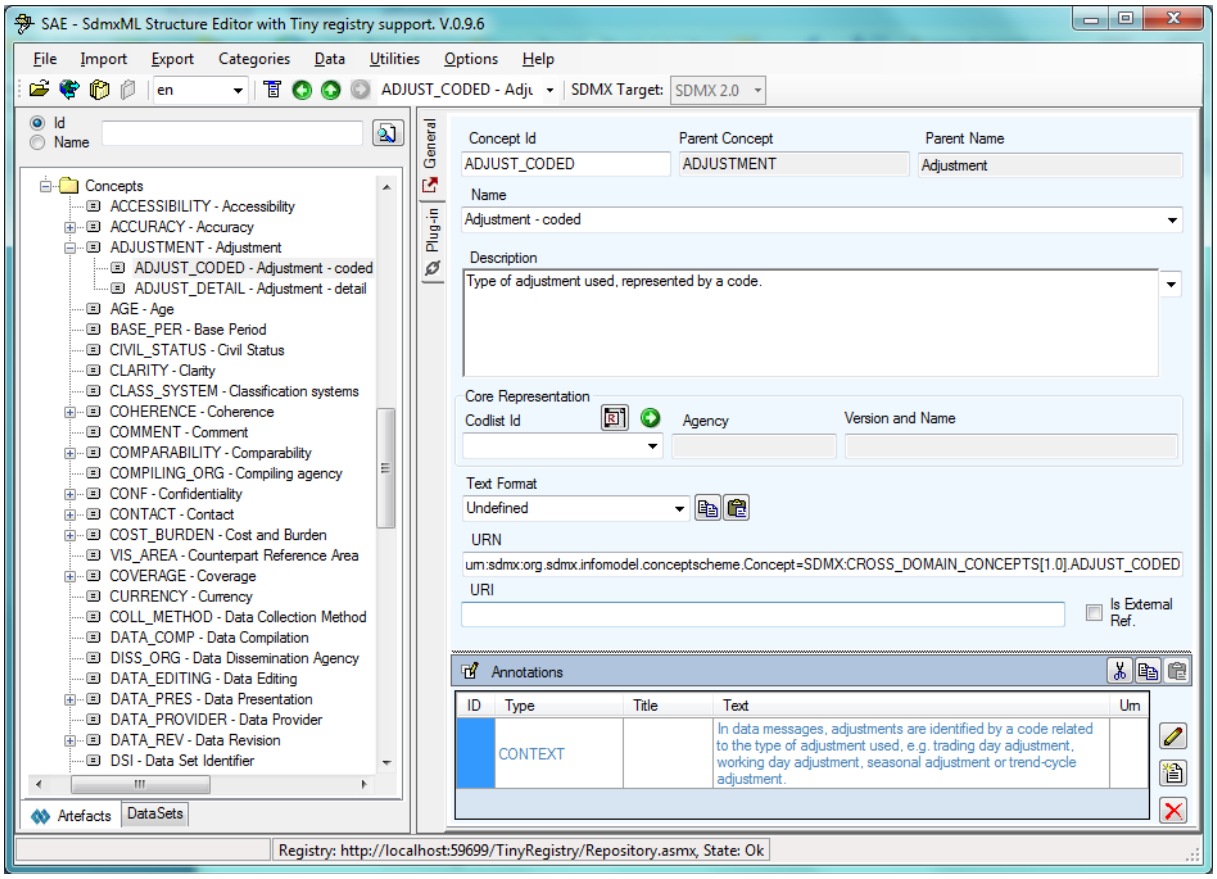
Export *ConceptScheme* information as PDF document. It includes associated code lists.

7.3.2 Stand Alone concepts

Note: Stand alone concepts are not managed by the application. If such an SDMX document is loaded, the global concepts will be grouped in a simulated scheme. This can create errors if global concepts of different agencies or version are in the document. See sample below for a simulated concept scheme. The references in DSD are also updated to concept scheme.



7.3.3 Concept detail



The detail screen manages the standard artefact attributes.Text and Description support multi language edition.
A text format can be specified. See “Text format paragraph for detail”.

A code list present in memory can be selected as core representation. The ESC key in combo box allows erasing any reference.

	Paste Code list urn. The core representation is given by pasting a previously copied codelist urn. .
	Navigation button. Navigate to in memory referred code list or loads it from repository if not yet present.

7.3.4 Concept hierarchy management

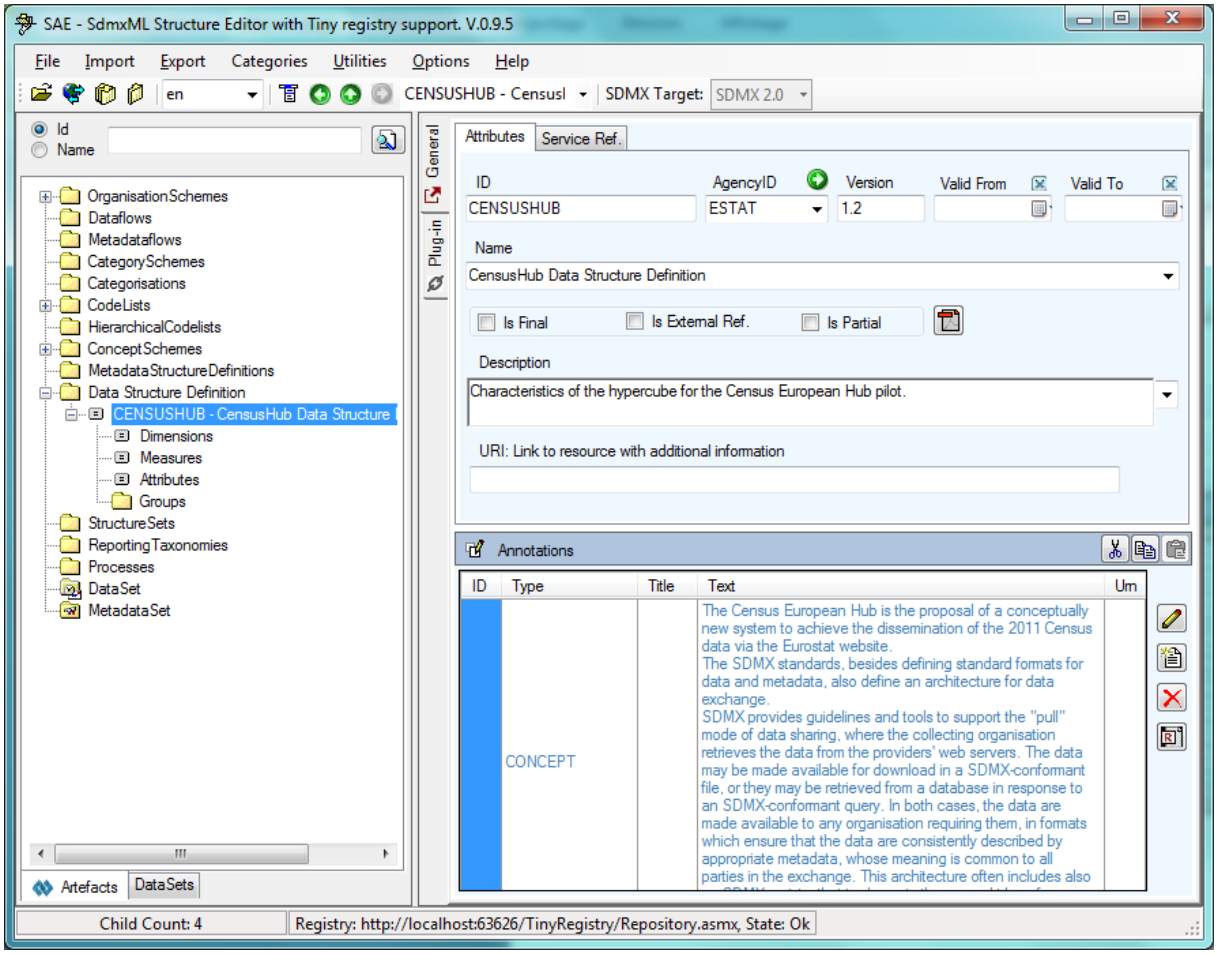
- The hierarchy of concepts is managed with the tree popup menu. Concept can be added as child of selected item and concept can be removed with all associated children.
- Parent hierarchy can also be redefined with popup menu functionality. See Code detail paragraph for explanation about hierarchy redefinition dialog box

- Concepts can be reordered (in current level) with drag and drop operation in tree.
- Finally concepts and scheme can be constructed with the Excel import functionality.

7.4 DSD

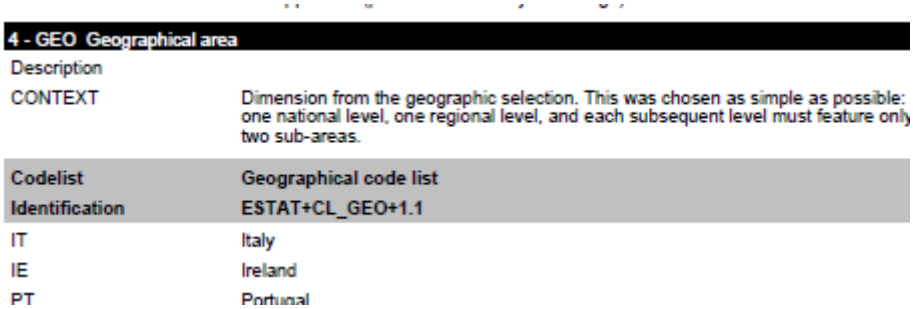
7.4.1 Artefact detail

The detail screen manages the standard artefact attributes.
Name and Description support multi language edition.



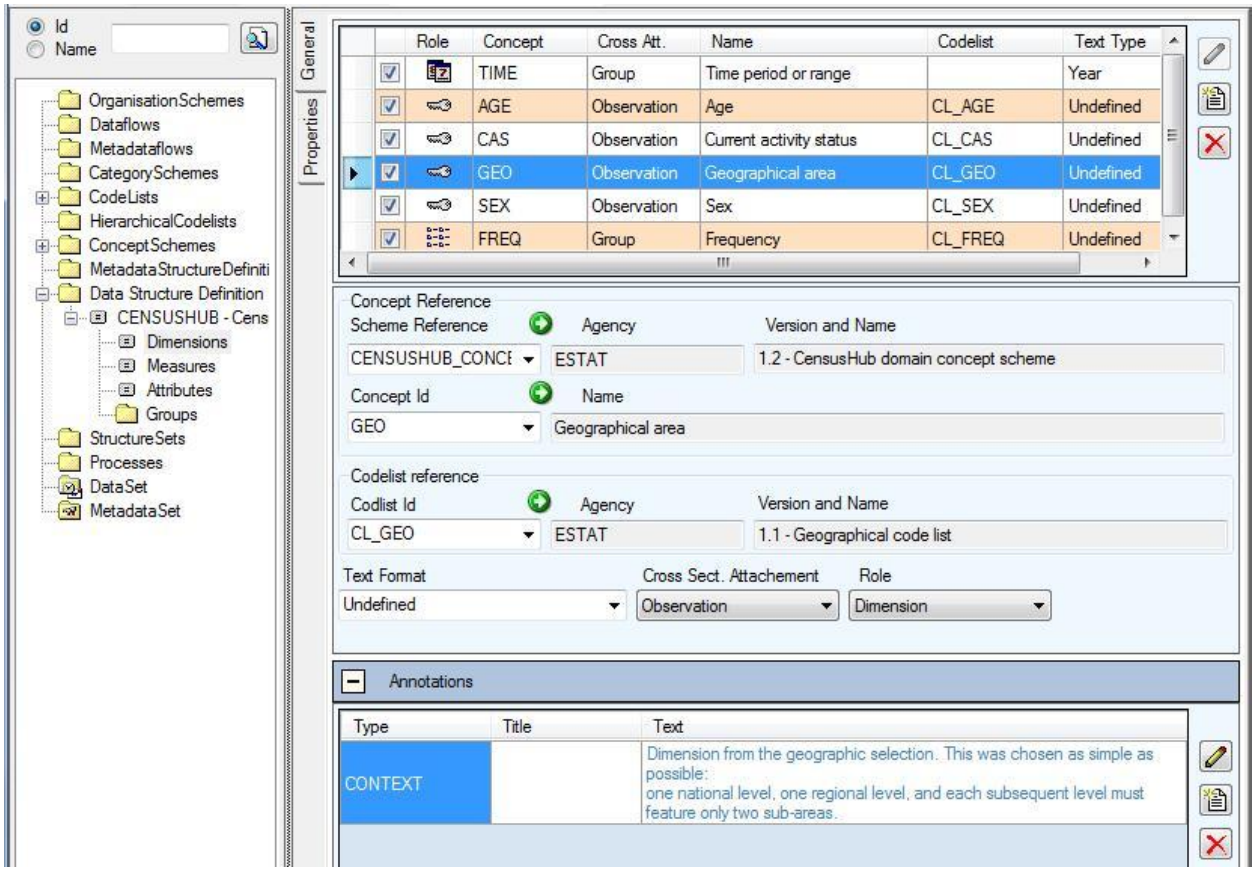
Export *DSD* information as PDF document. It includes associated code lists And concepts information..

Extract of PDF export.



4 - GEO Geographical area	
Description	
CONTEXT	Dimension from the geographic selection. This was chosen as simple as possible: one national level, one regional level, and each subsequent level must feature only two sub-areas.
Codelist	**Geographical code list**
Identification	**ESTAT+CL_GEO+1.1**
IT	Italy
IE	Ireland
PT	Portugal

7.4.2 Dimensions



The dimensions are grouped in a data grid view. The selected row of the grid can be edited with the controls of the middle panel. The annotation(s) associated to the selected dimension are reflected in the annotation panel. The dimensions can be reordered with drag and drop function in the grid. It is the user responsibility to order them correctly. The time dimension is not part of the collection of dimension in the SDMX document, so its order is meaningless. The vertical size of the grid can be redefined by the horizontal split bar located below the grid area.

Navigation to referenced elements is possible if these artefacts are in memory.

Only one time dimension can be created. When none are defined, an additional button is displayed in grid area.

Grid functions

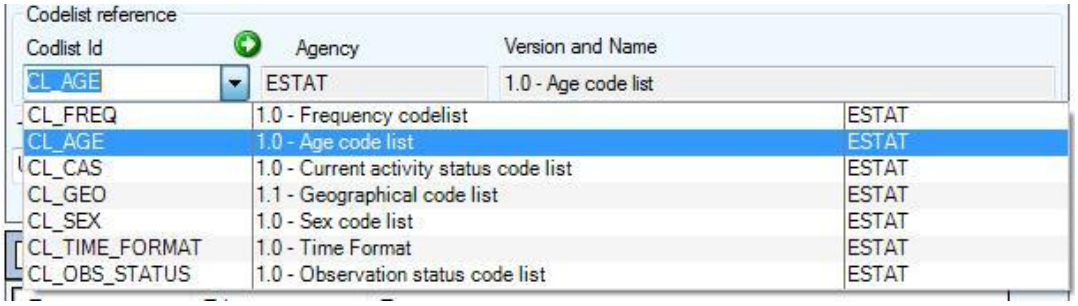
	Unused, selected row can automatically be edited in middle panel.
	Creates a new dimension.
	Deletes selected dimension.
	Creates the time dimension. It is visible only if no time dimension defined.

Middle Panel

References to concept and to code list can be done with artefacts loaded in memory.

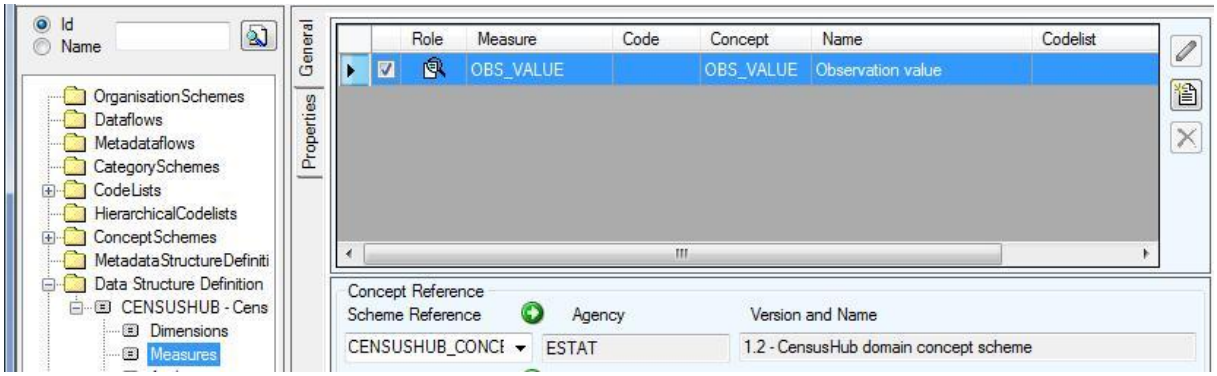
Others combo boxes allow to characterize the dimension.

Sample of reference edition


Codelist reference
Codlist Id
Agency
Version and Name
CL_AGE
ESTAT
1.0 - Age code list
CL_FREQ 1.0 - Frequency codelist ESTAT
CL_AGE 1.0 - Age code list ESTAT
CL_CAS 1.0 - Current activity status code list ESTAT
CL_GEO 1.1 - Geographical code list ESTAT
CL_SEX 1.0 - Sex code list ESTAT
CL_TIME_FORMAT 1.0 - Time Format ESTAT
CL_OBS_STATUS 1.0 - Observation status code list ESTAT

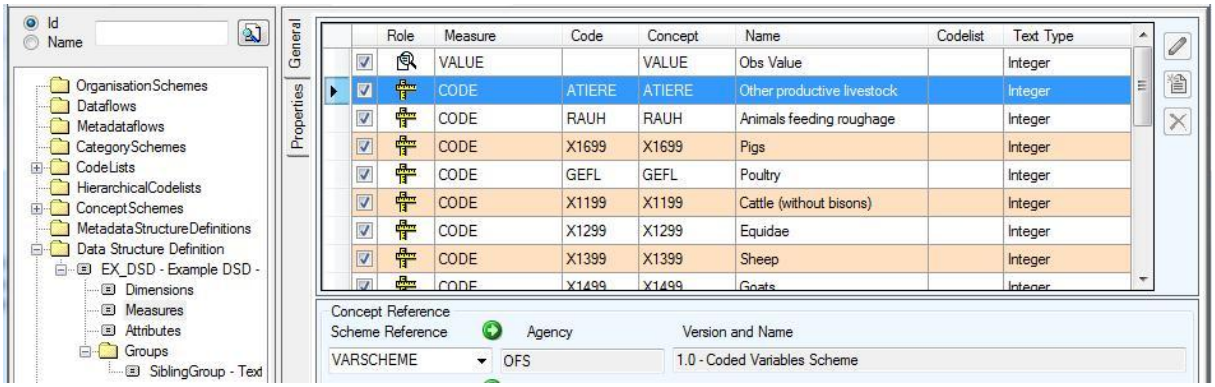
7.4.3 Measures

Only Primary measure



The measure presents the primary measure in the grid as well as the cross sectional measures (if any)

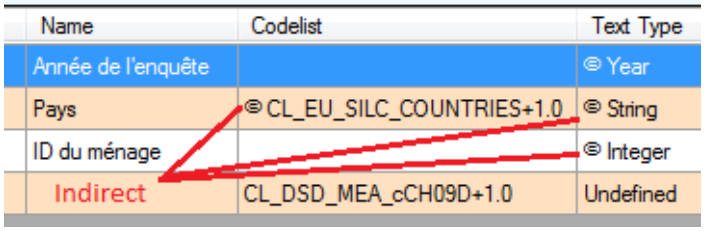
Cross sectional measures



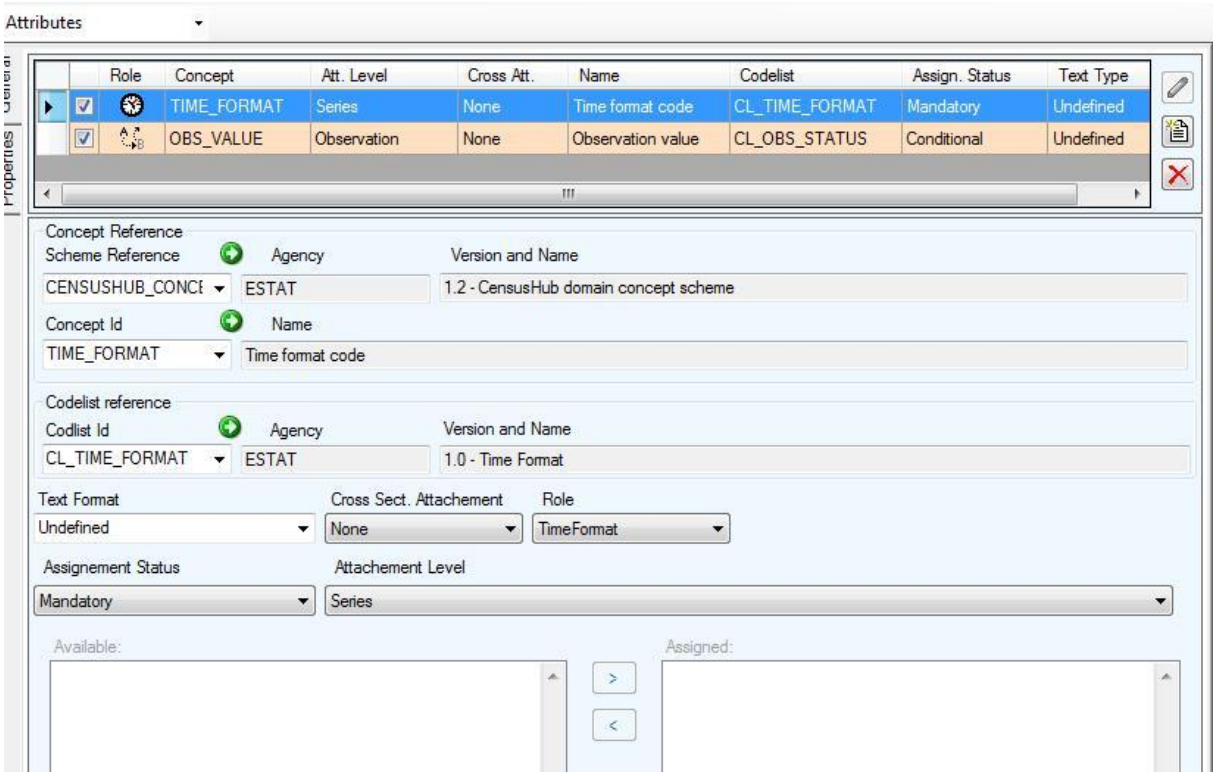
The set of cross sectional measures is no more created in one shot according to codelist(s) defined in measure dimension(s), it is now managed independently.

Distinguish direct references and references given through concept

If a reference to codelist or a text format is not defined in DSD, the corresponding information (if any) given in concept are used instead. In order to highlight this, the grid display differs slightly for each case. Indirect references are preceded by a marker indicating coming from concept:



7.4.4 Attributes



The attributes are grouped in a data grid view. The selected row of the grid can be edited with the controls of the middle panel. The annotation(s) associated to the selected attribute are reflected in the annotation panel. The attributes can be reordered with drag and drop function in the grid.

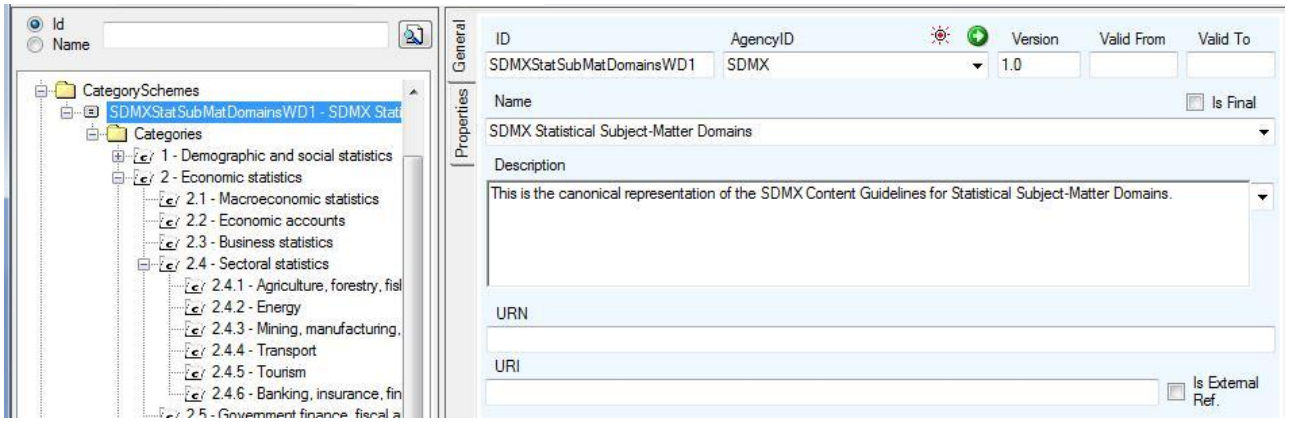
For **group attachment**, the group to use can be selected in the attachment dual box. (Available, Assigned).

This dual box is also used for **observation level attachment**.

7.5 Category scheme

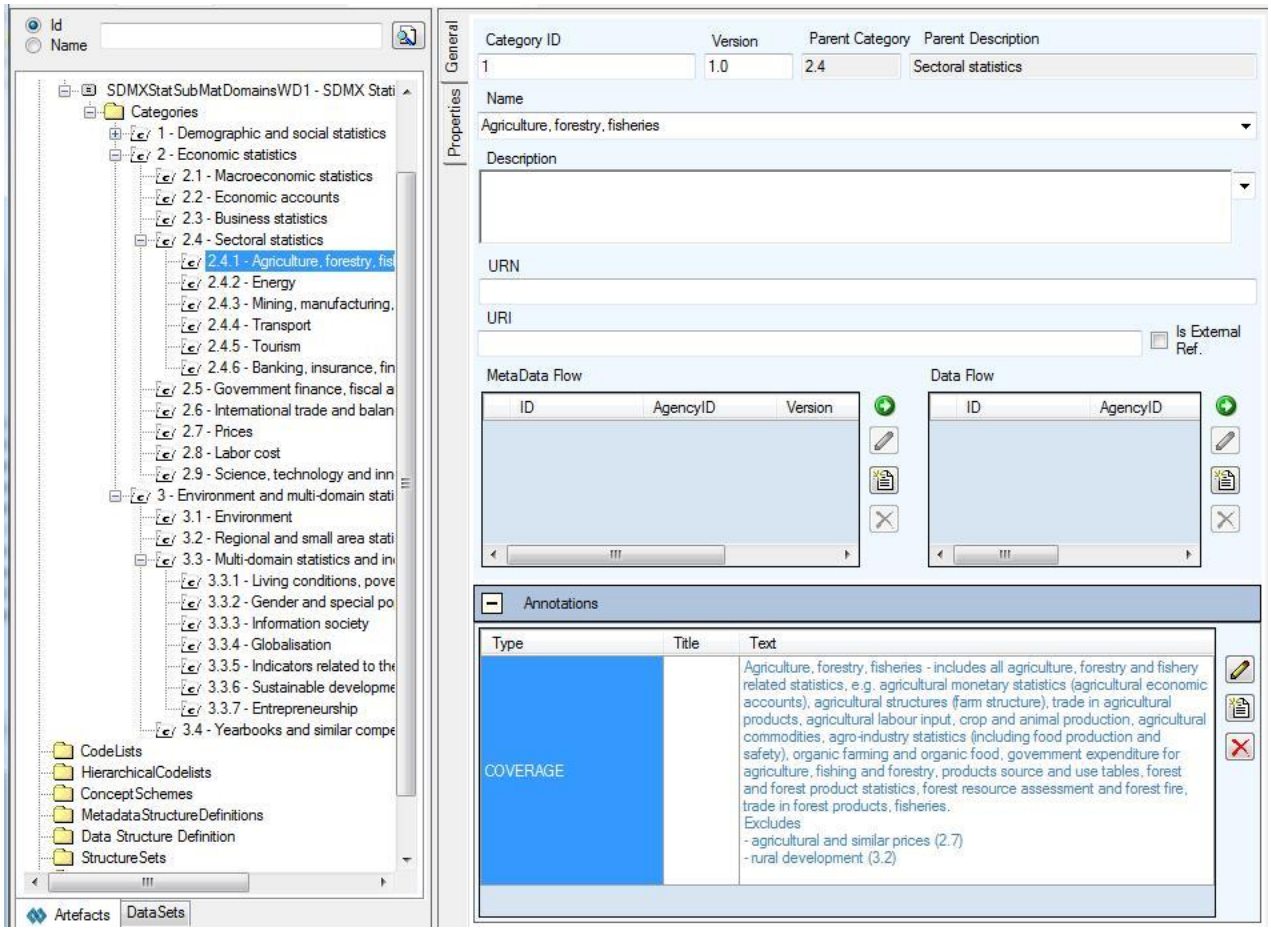
7.5.1 Artefact detail

The detail screen manages the standard artefact attributes.
Name and Description support multi language edition.



7.5.2 Category detail

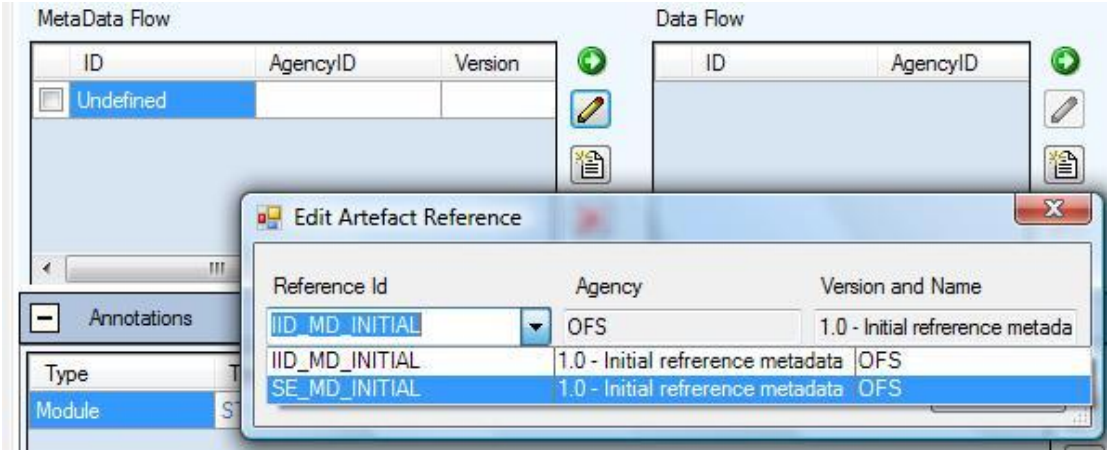
The Category hierarchy is a different hierarchy than code list or concepts. It allows same id to occur in different levels of the hierarchy. The idea is to compose identification by grouping all levels id. In sample screen, the id of third level is “1” but its representation is “2.4.1”. The tree view reflects this convention.



The detail screen manages the standard artefact attributes. Text and Description support multi language edition.

Flow Attachment

Data flows and metadata flows can be attached with the flow grid buttons.



	Edit selected reference in flow grid. See reference editor dialog box.
	Creates a new flow reference as a new grid row.
	Deletes selected reference row in grid.
	Navigate to artefact according to reference given in flow grid row. Artefact must be loaded in memory.

7.5.3 Category hierarchy management

- The hierarchy of category is managed with the tree popup menu. Category can be added as child of selected item and can be removed with all associated children.
- Parent hierarchy can also be redefined with popup menu functionality. As the hierarchy allows same id to occur in different levels, it is the user responsibility to redefine eventual duplicated id after a parent redefining. The full identification is used for display.

  See Code detail paragraph for explanation about hierarchy redefinition dialog box
- Categories can be reordered (in current level) with drag and drop operation in tree.
- Finally category scheme can be constructed with the Excel import functionality.

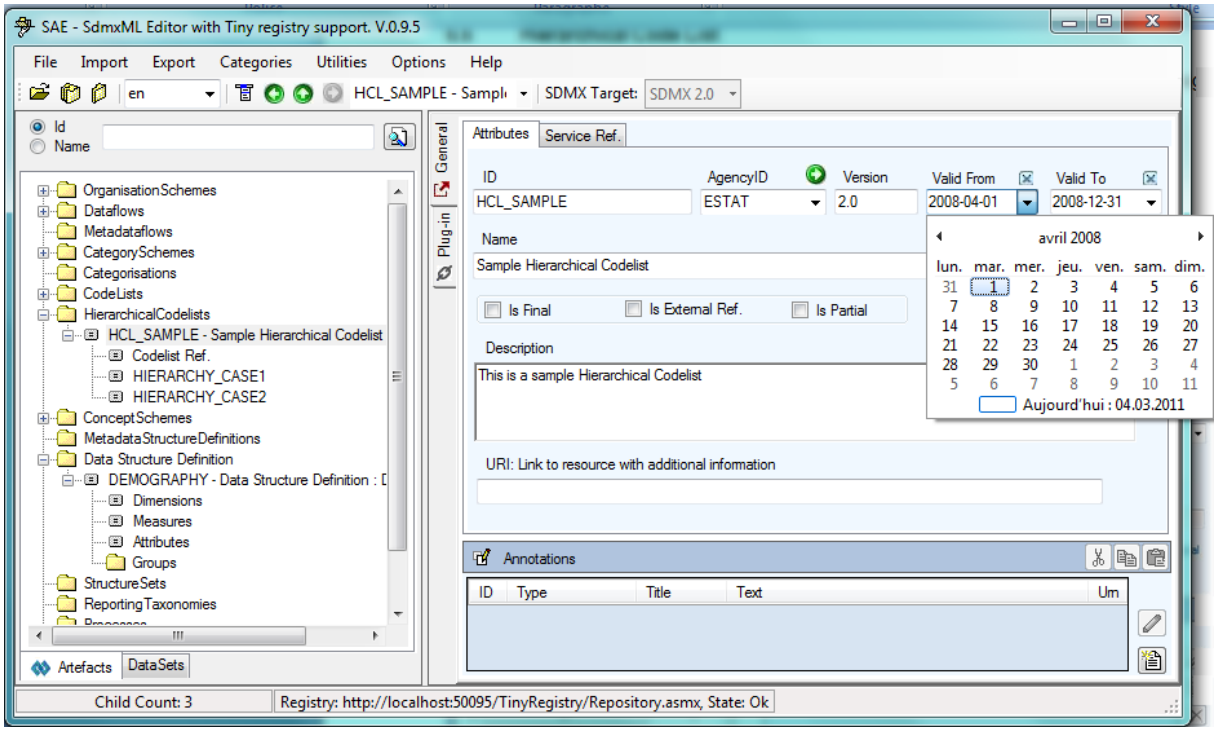
Note: Some SDMX category artefact documents do not use the “Composed identification” convention in categories. The application allows it but the tree display is a bit clumsy.

7.6 Hierarchical Code List

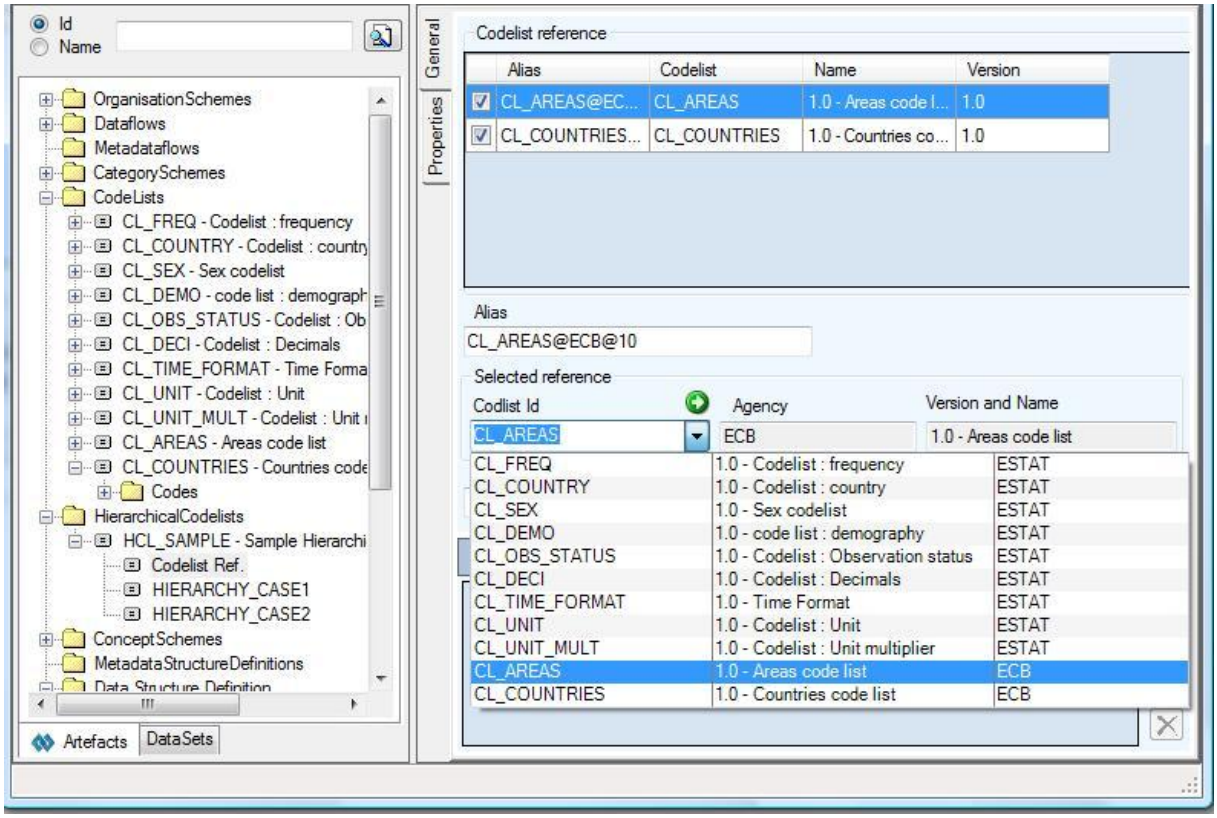
Hierarchical code list allows defining variant hierarchy on codes. It also allows managing history of codes validity.

7.6.1 Artefact detail

The detail screen manages the standard artefact attributes.
Name and Description support multi language edition.
The URI box allows a link to additional information. When filled, a navigation button is available. The checkbox "*Is Partial*" has no meaning in SDMX 2.0



7.6.2 Code lists reference



References can be selected from the set of in memory code list artefacts.
Navigation to referred code list can be done with navigation button.

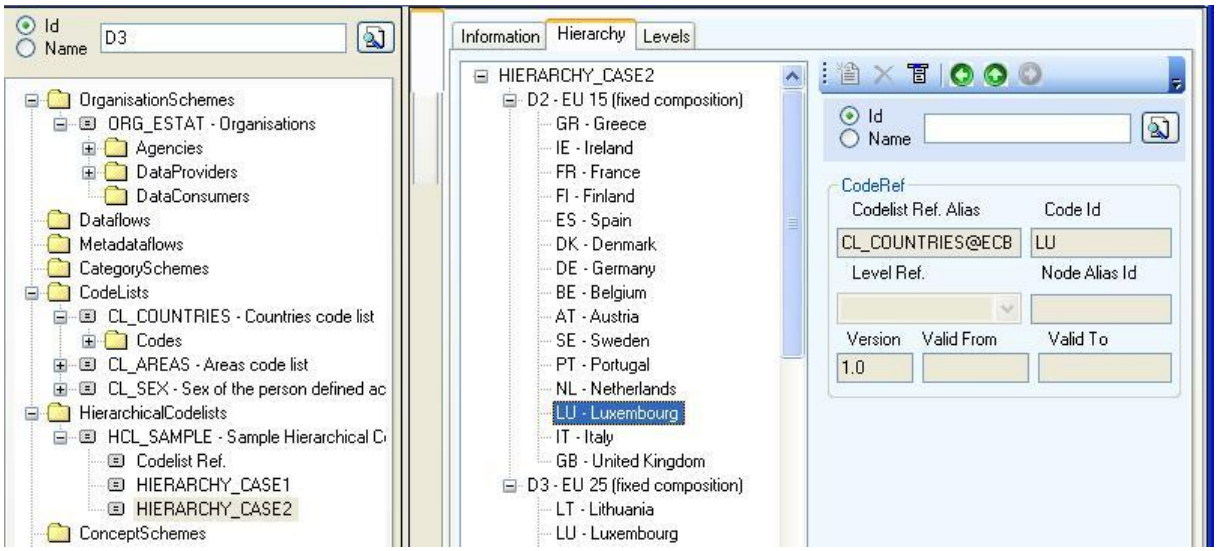
Hierarchy details are presented in a tabbed detail view composed of three panels.

7.6.3 Information Tab details

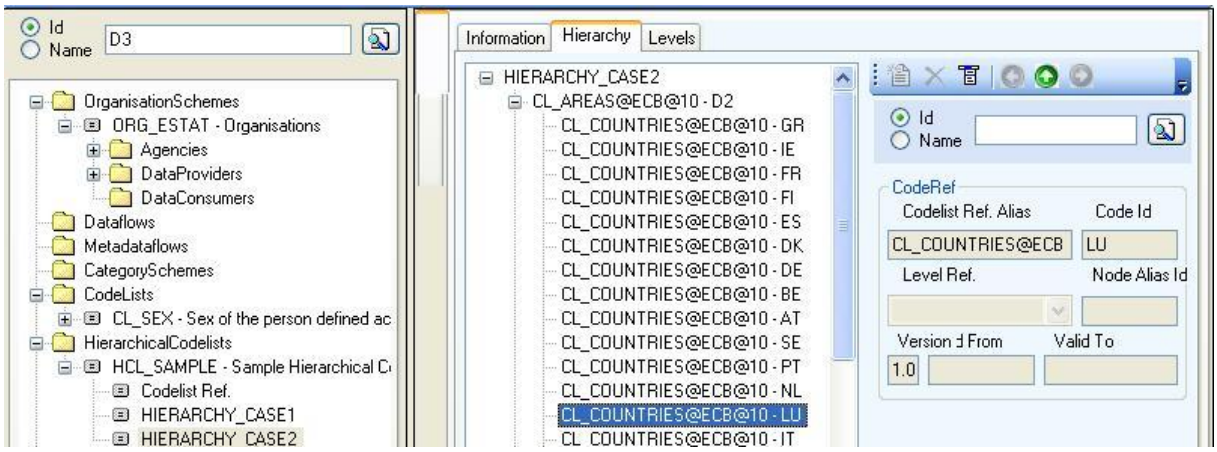
The first panel regroups hierarchy information attributes, more or less similar to artefact attributes detail view. Name and Description support multi language edition.

7.6.4 Hierarchy Tab details

This panel displays a tree representation of the defined hierarchy. On the left side of the hierarchy tree, a set of controls allows to edit the selected item in the hierarchy tree.

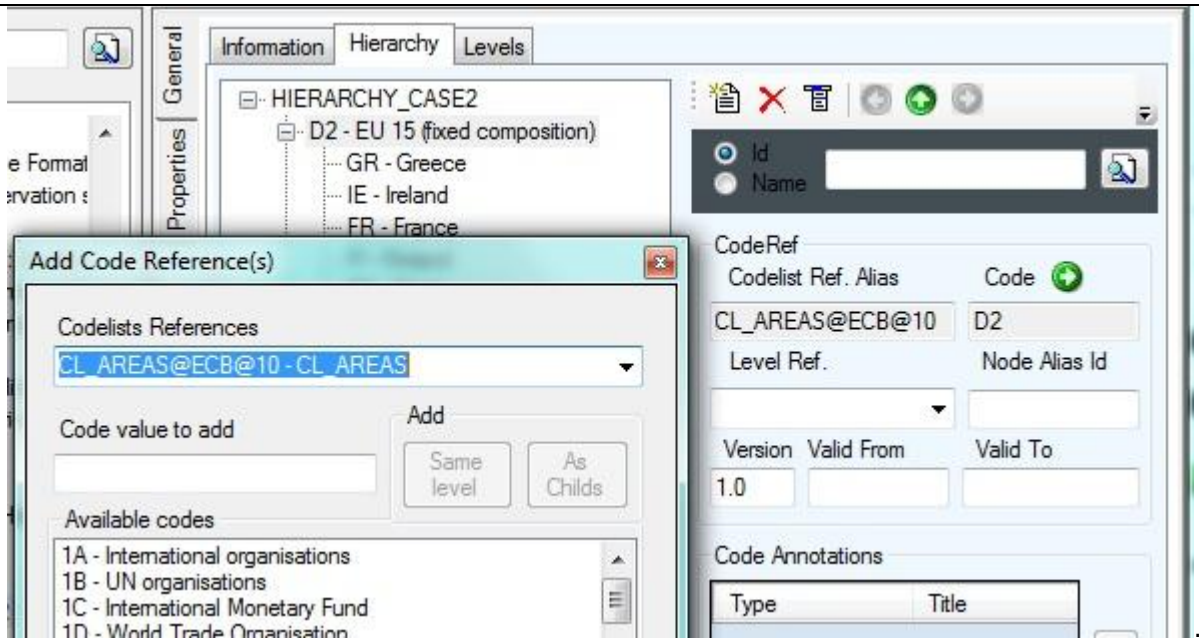


If the referred code lists are not loaded in memory, the hierarchy tree will only displays code list reference and code value.



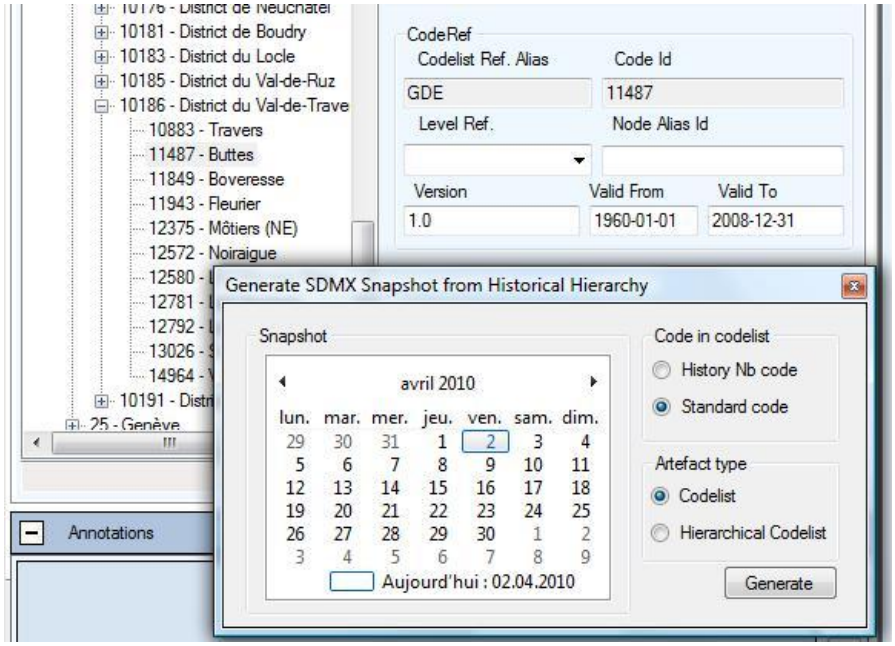
The detail view contains also a tool bar that regroups functions to operate on the hierarchy.

	Deletes selected code in the tree hierarchy. All children items are also removed.
	Navigation bar functions, as in main tool bar, but it operates on hierarchy tree,
	This function is only available if referred code lists are loaded in memory. It allows attaching some codes in the hierarchy according to the selected item. This function opens a dialog box where user can select code list between referred ones and to select a subset of codes to attach in the hierarchy. Codes can be attached to the same level of the tree selected code or as children of the selected code in hierarchy tree.



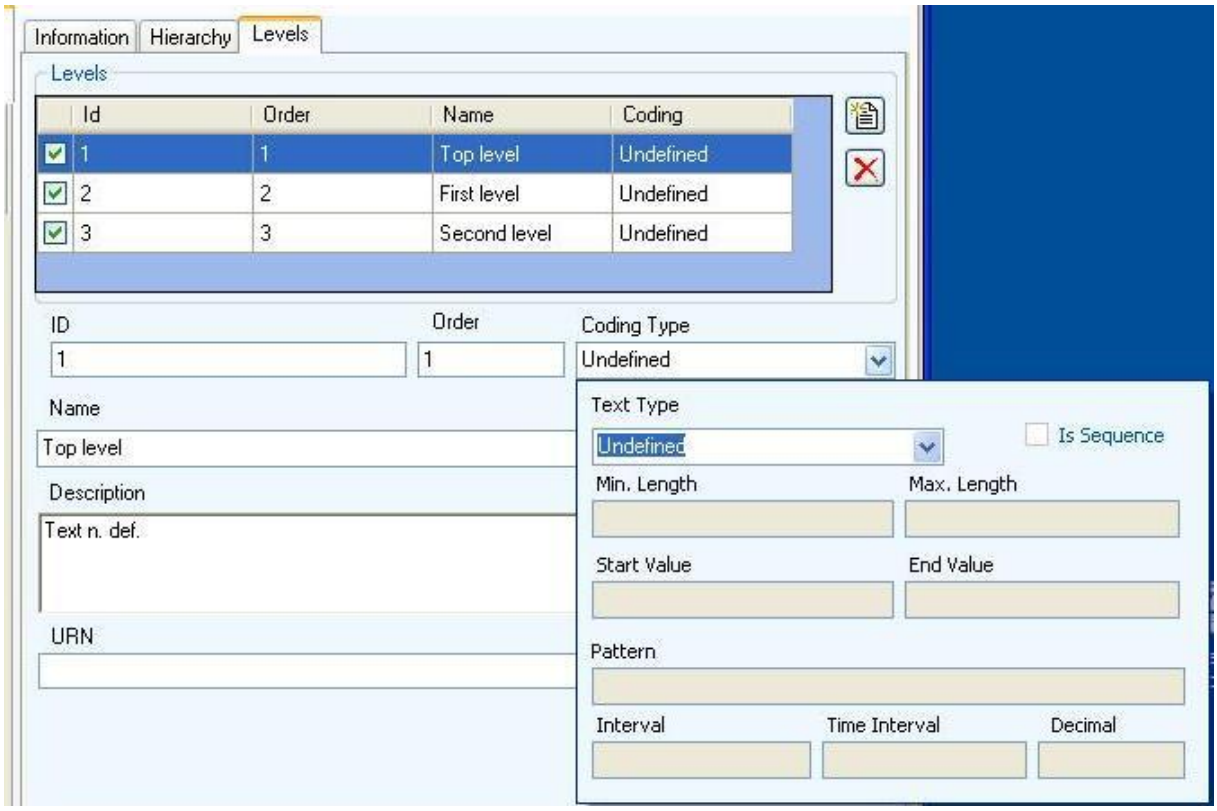
Specialized function that can be used if the hierarchy is managed as historical, by using a history code in code list codes and by using valid from and valid to attributes in the code reference.

This function creates a SDMX artefact that is a snapshot of the codes hierarchy that are valid at the specified time.



7.6.5 Level Tab details

This panel displays The Levels definition.



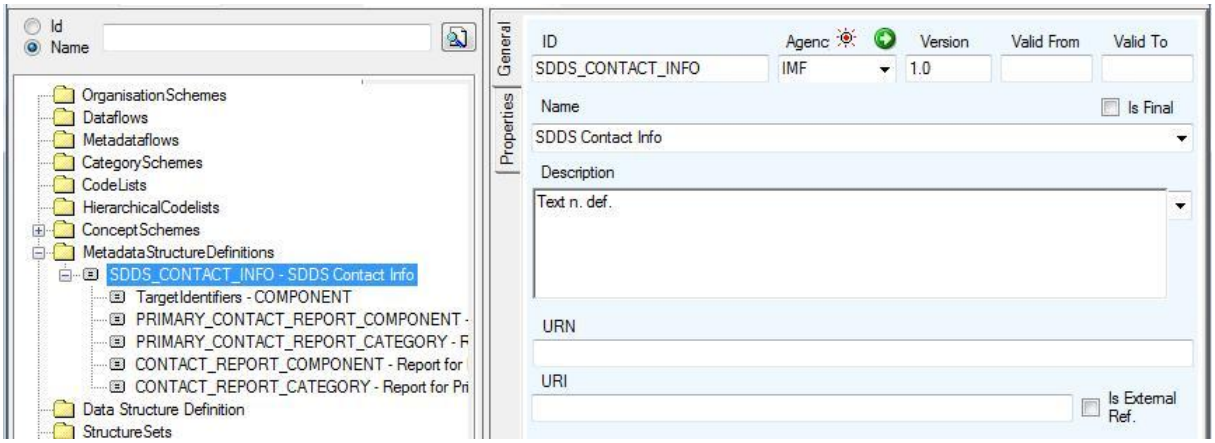
The central panel allows editing the level selected in the levels grid. Name and Description support multi language edition.

The coding type edit control is presented in an activated state. It allows specifying details about coding type.

7.7 MSD

7.7.1 Artefact detail

The detail screen manages the standard artefact attributes.
Name and Description support multi language edition.
All defined reports are presented as child of the MDS as well as target identifiers information.



7.7.2 Target Identifiers

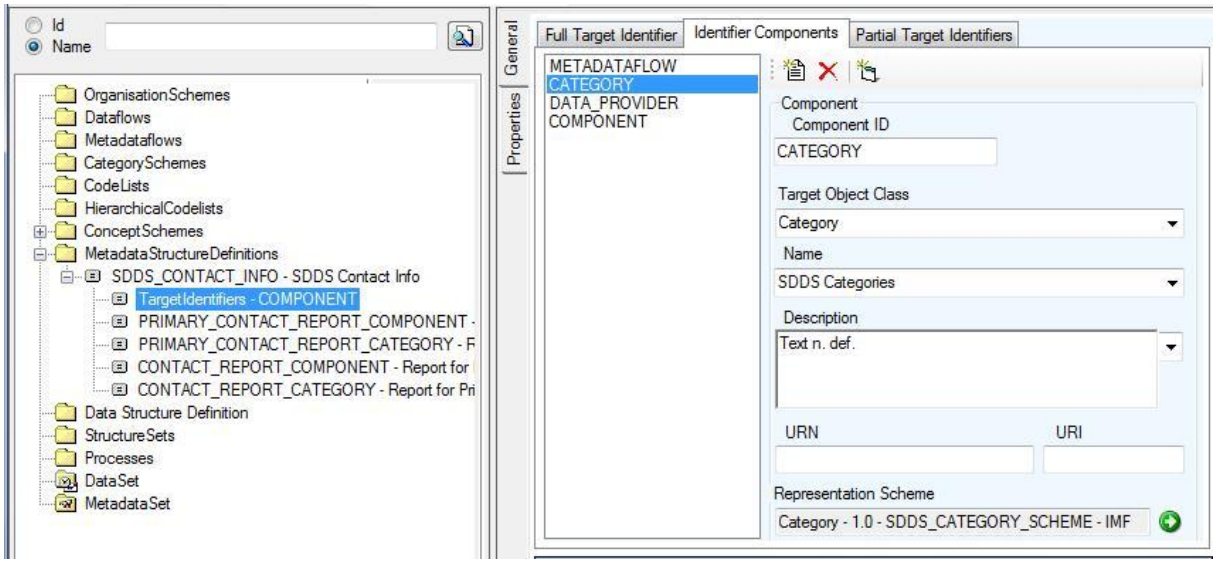
The target identifiers detail view is composed of three panels.

Full target identifier Tab

The first panel regroups full target identifier information attributes, more or less similar to artefact attributes detail view. Name and Description support multi language edition.

Identifier components Tab

This panel is used to define the full target identifier keys.

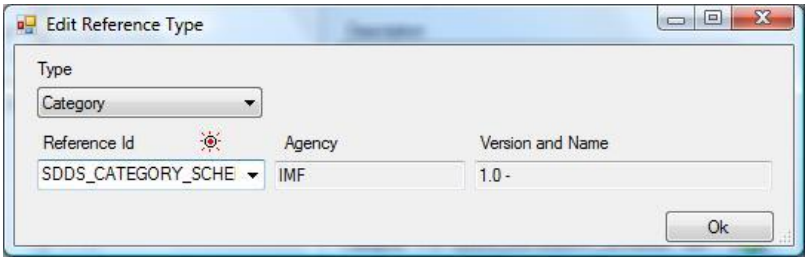


- The defined keys are listed on the left of the detail view. The right area is used to edit detail of selected key in list. Name and Description support multi language edition.

- The items in the list can be reordered through drag and drop operation.

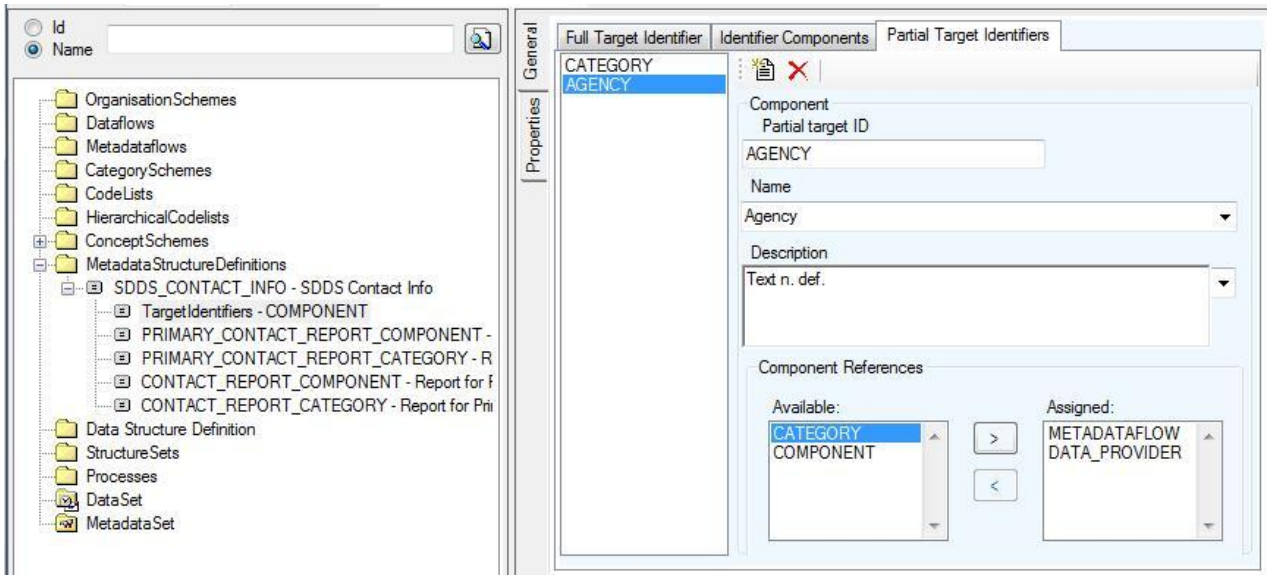
A tool bar is available to operate on full targets.

	Creates a new key in identifiers list.
	Deletes selected key in identifiers list.
	Raises a dialog box to edit the Representation scheme associated to the identifier.
	Navigates to representation scheme artefact according to reference given. Artefact must be loaded in memory.



Partial target Identifier Tab

This panel is used to define the partial target identifiers. These are based on full identifiers.



The partial keys defined are listed on the left of the detail view. The right area is used to edit detail of selected key in list.

A dual list is available to assign the subset of full target identifiers associated to the partial key. The dual list reflects the available identifier components.

A tool bar is available to operate on partial targets.

	Creates a new key in partial identifiers list.
	Deletes selected key in partial identifiers list.

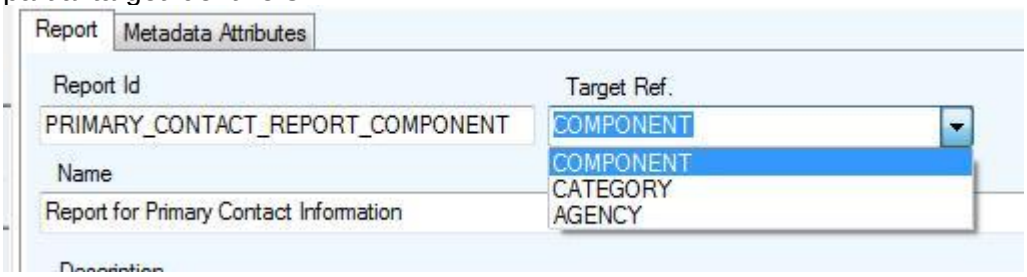
7.7.3 Report

The target Report detail view is composed of two panels.

Report Tab

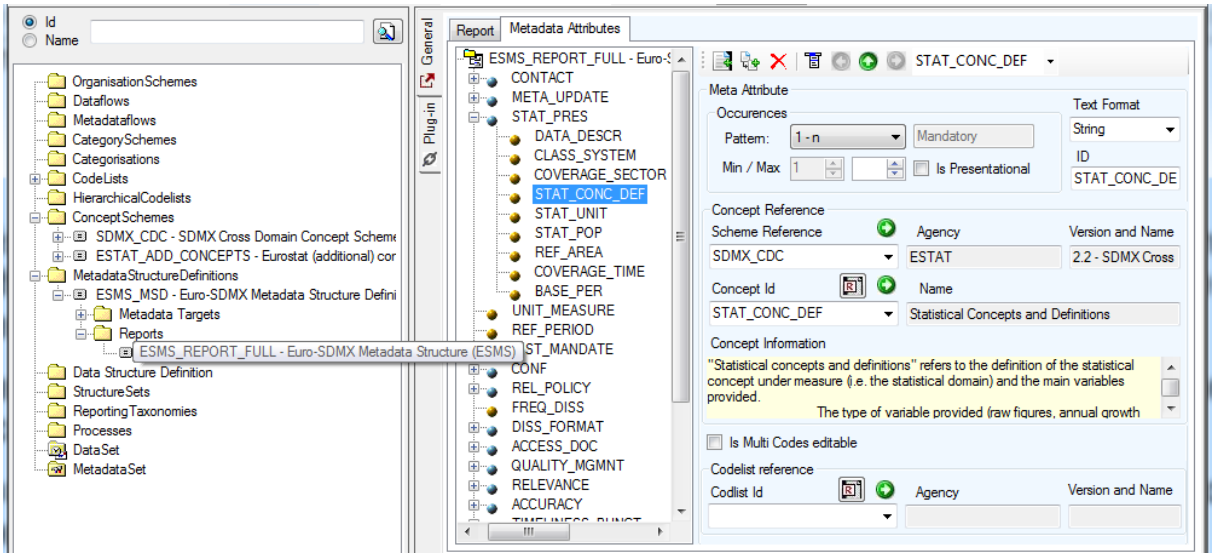
The first panel regroups report information attributes, more or less similar to artefact attributes detail view. Name and Description support multi language edition.

The target reference combo box allows selecting between full identifier or one of the optional partial target identifiers.



Meta data attributes Tab

This panel is used to define the hierarchy of metadata attributes based on concepts selected from relevant concept schemes.



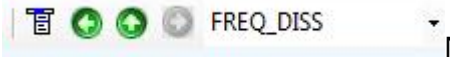
This panel displays a tree representation of the defined concepts hierarchy. On the left side of the hierarchy tree, a set of controls allows to edit the selected attribute in the hierarchy tree.

The text format control can be used to specify details about the attribute text format. According to the usage defined, the tree reflects a different icon colour.

A concept reference is associated to the attribute. An optional code list reference can also be given. References on memory loaded artefacts can be edited.

A tool bar is available to operate on attributes.

	Creates a new attribute after the selected item in the attributes hierarchy tree.
	Creates a new attribute as child of the selected item in the attributes hierarchy tree.
	Deletes selected attribute and all its children. A confirmation dialog box is used.



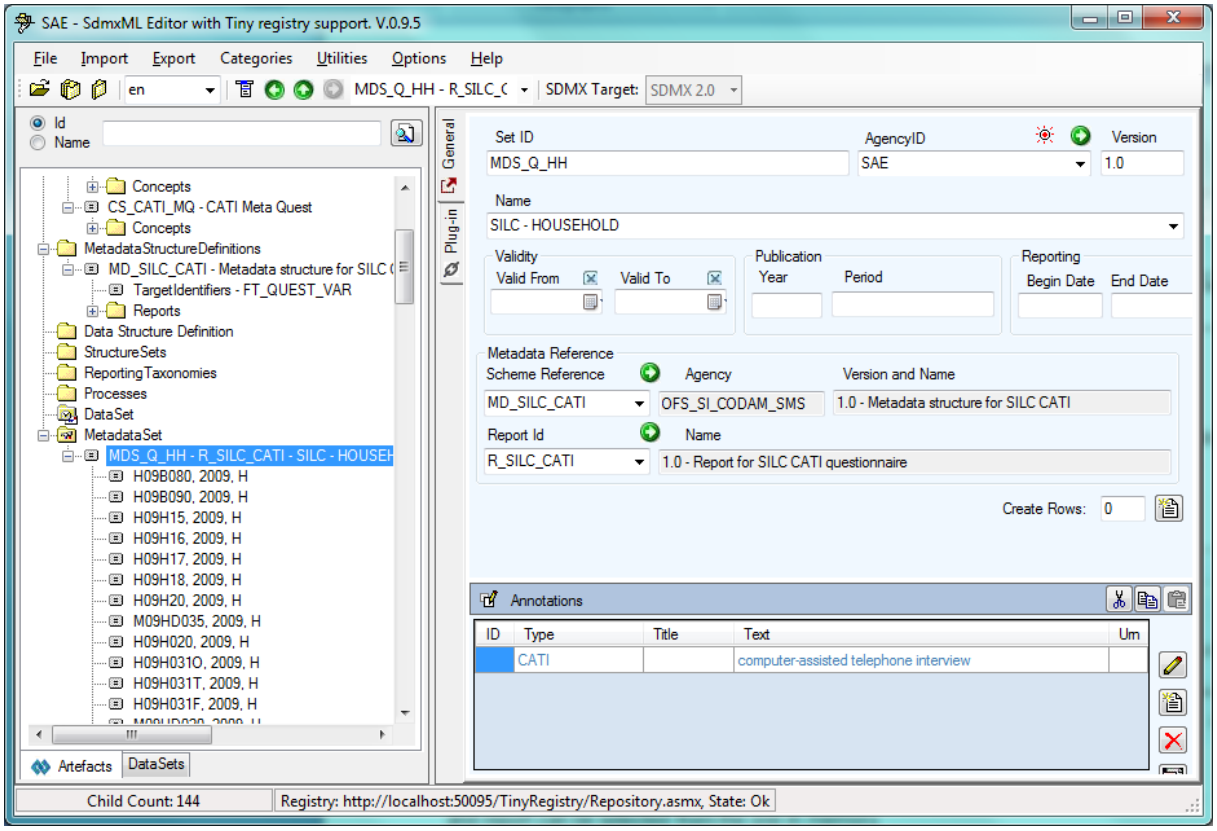
Navigation toolbar functions operate on attribute tree.

	Navigates to referenced artefact according to reference given. Artefact must be loaded in memory.

7.8 Metadata set

The Application reads and allows creation of metadata set information. For creating metadata set rows, the referred metadata structure and report need to be loaded in memory. The metadataset document should be of type “generic metadataset”

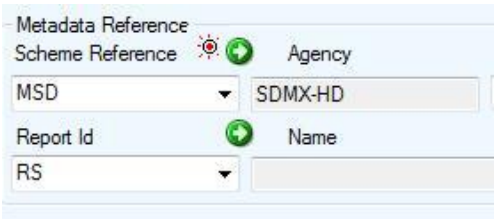
7.8.1 Artefact detail



The Set Id, Agency Id and version are not part of the SDMX 2.0 standard; they are added to provide enough information to the TinyRegistry for managing “generic metadata” document. This information is stored as an annotation of type “***SetID***” in the document.

An internationalized title name can be given and reference to metadata structure definition and report can be selected from those in memory.

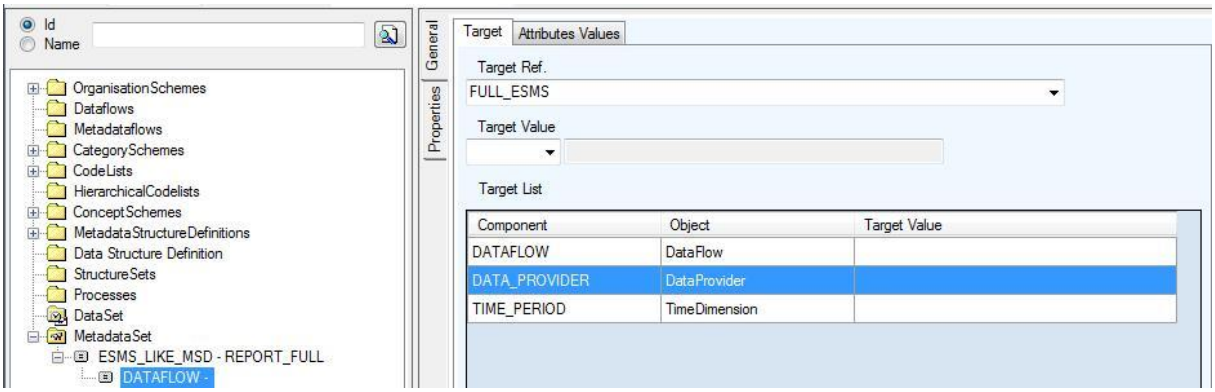
If metadata structure is not available in memory, a red icon signals it.



Rows of metadata can then be created by the popup menu “new child” or by the create row function button.

New rows can only be created if the metadata structure and report information is available in memory.

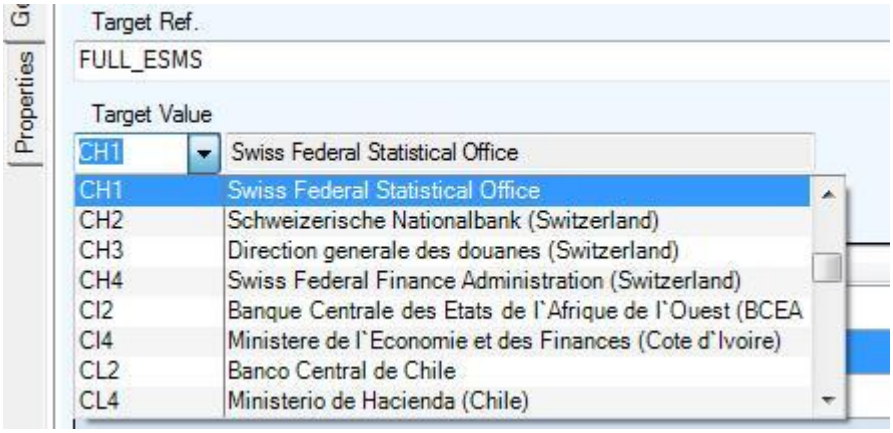
7.8.2 Artefact row detail



The Row detail screen is composed of two tabs. The first one groups the target keys values defined to identify the row. During row creation, the Target ref is automatically added according to associated report information.

The data grid displays the key values and the selected row in grid is editable via the target ref control. This control is presented as a drop down if a code list or a category is associated to the key. Otherwise it is a simple text box.

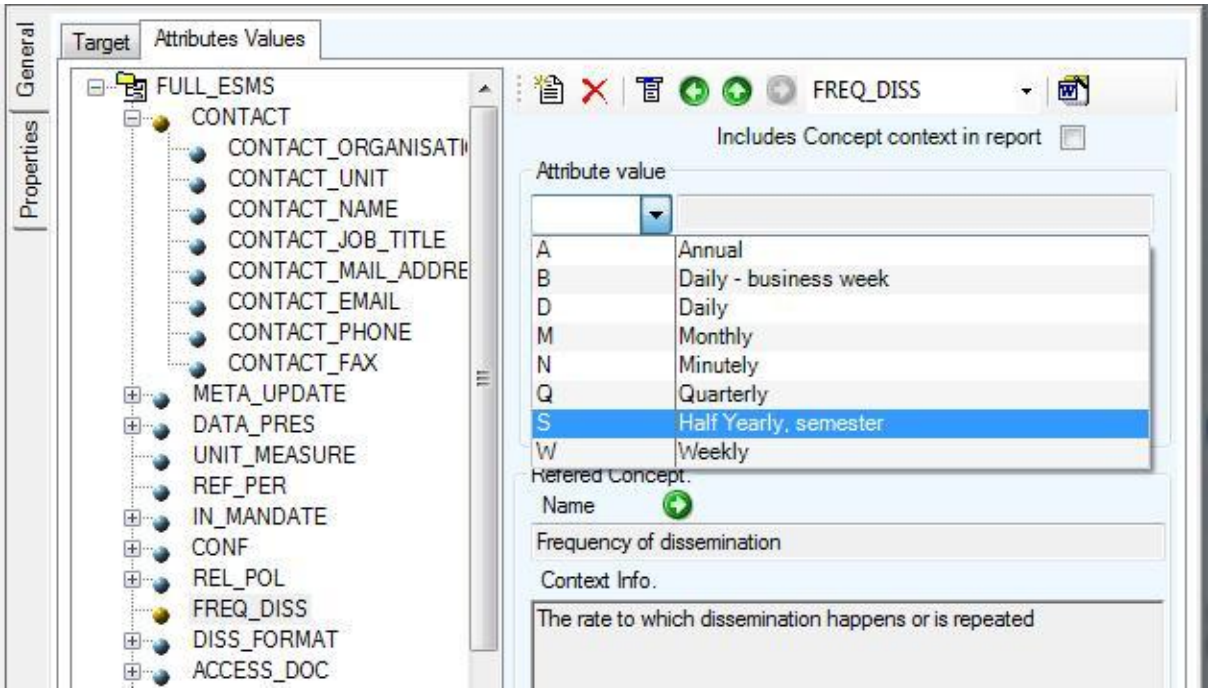
See below the sample Target value selection for the Data provider key.


Properties
Target Ref.
FULL_ESMS
Target Value
CH1
Swiss Federal Statistical Office
CH1 Swiss Federal Statistical Office
CH2 Schweizerische Nationalbank (Switzerland)
CH3 Direction generale des douanes (Switzerland)
CH4 Swiss Federal Finance Administration (Switzerland)
CI2 Banque Centrale des Etats de l`Afrique de l`Ouest (BCEA
CI4 Ministere de l`Economie et des Finances (Cote d`Ivoire)
CL2 Banco Central de Chile
CL4 Ministerio de Hacienda (Chile)

7.8.3 Row Attributes values

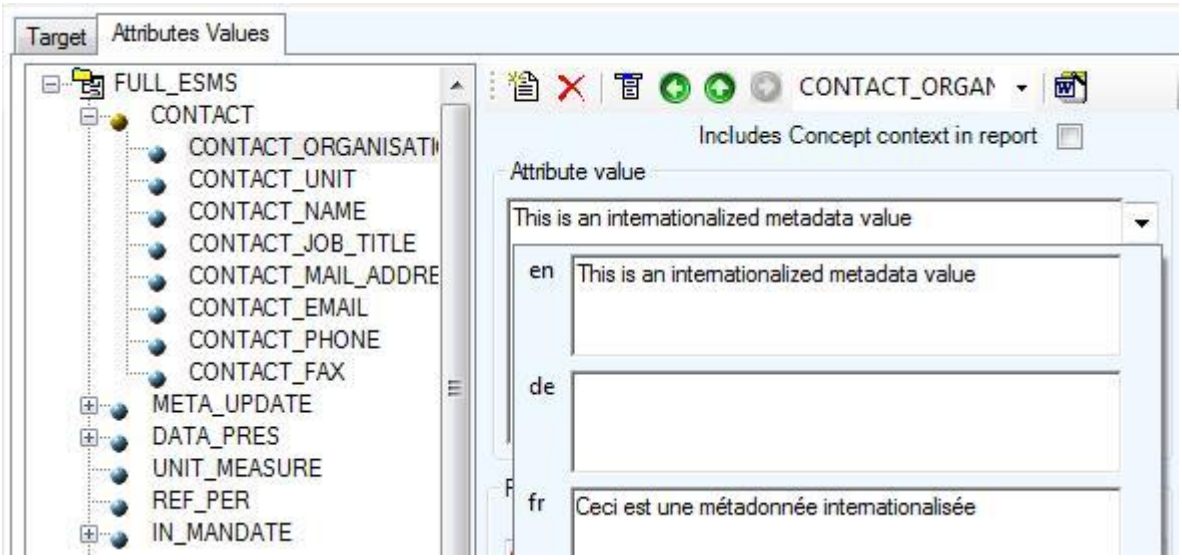
The second tab presents a tree view with the defined concepts associated to attributes values. According to selected tree item, concept name, description and eventual context annotation are displayed as information. A tool bar is available to operate on attributes.

	Remove all non filled, non mandatory attributes of the row. A confirmation dialog box is used.
	Append all missing concepts of the metadata row according to report information.
	Navigation toolbar functions operate on attribute tree.
	Generate a sample Word report. This illustrates the possibility to use transformation like XSLT to generate HTML documents from metadata set. If the checkbox “Includes concepts info” is checked, concept context info are added.



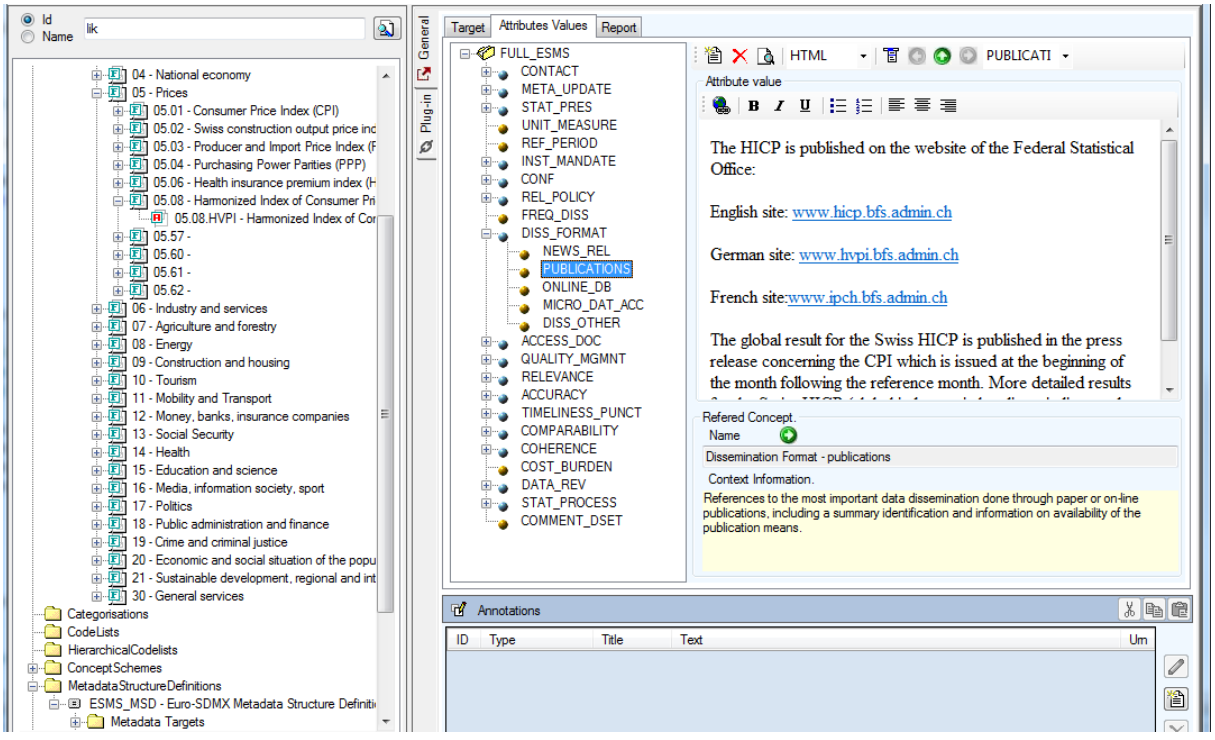
According to the associated concept text type, the attribute value can be selected from a list, edited in different languages or written in a simple numeric or text box.

See below sample for an internationalized attribute value.



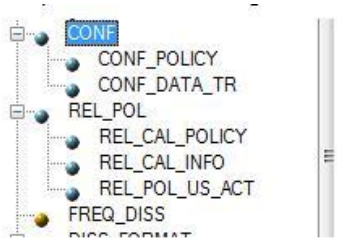
7.8.4 report of metadata embedding HTML.

A specialized view can be activated to display/edit attribute values that uses embedded HTML. See below sample of an item of an ESMS report.



7.8.5 Sample report of metadata set row.

The image below illustrates a part of a sample report.



CONF

CONF_POLICY

431.01 Loi du 9 octobre 1992 sur la statistique fédérale (LSF) 431.011 Ordonnance du 30 juin 1993 concernant l'organisation de la statistique fédérale 431.012.1 Ordonnance du 30 juin 1993 concernant l'exécution des relevés statistiques fédéraux (Ordonnance sur les relevés statistiques)

CONF_DATA_TR

Religions: niveau de protection des données 3 (données personnelles sensibles) Contrat de protection des données avec mandataires externes

REL_POLICY

REL_CAL_POLICY

Au plus tard 1 an après l'enquête

REL_CAL_ACCESS

Disponible (à préciser)

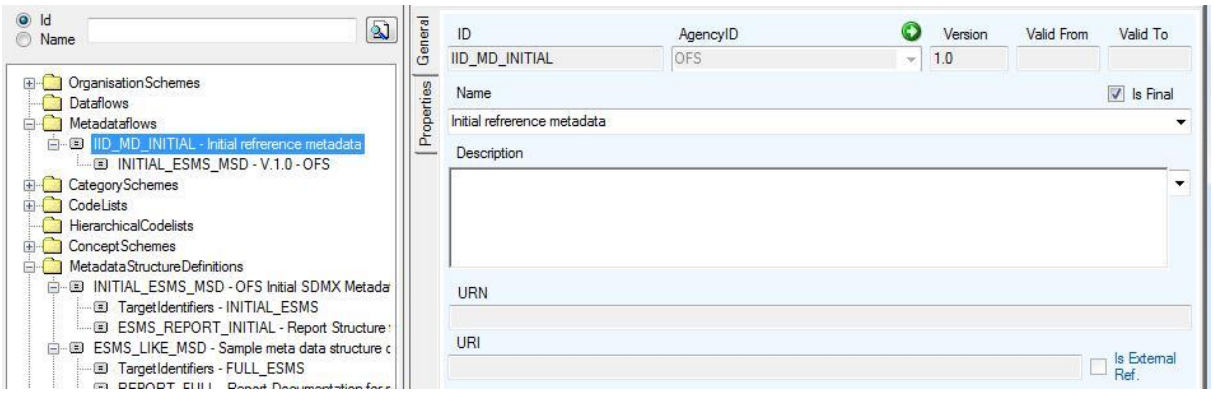
REL_POL_US_AC

Tous les canaux prévus par DIAM (à préciser)

FREQ_DISS

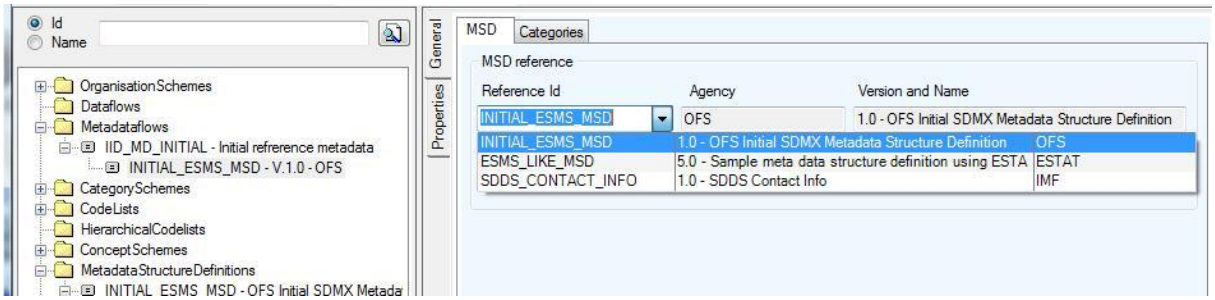
7.9 Data flow and Metadata flow

7.9.1 Artefact detail



7.9.2 Reference

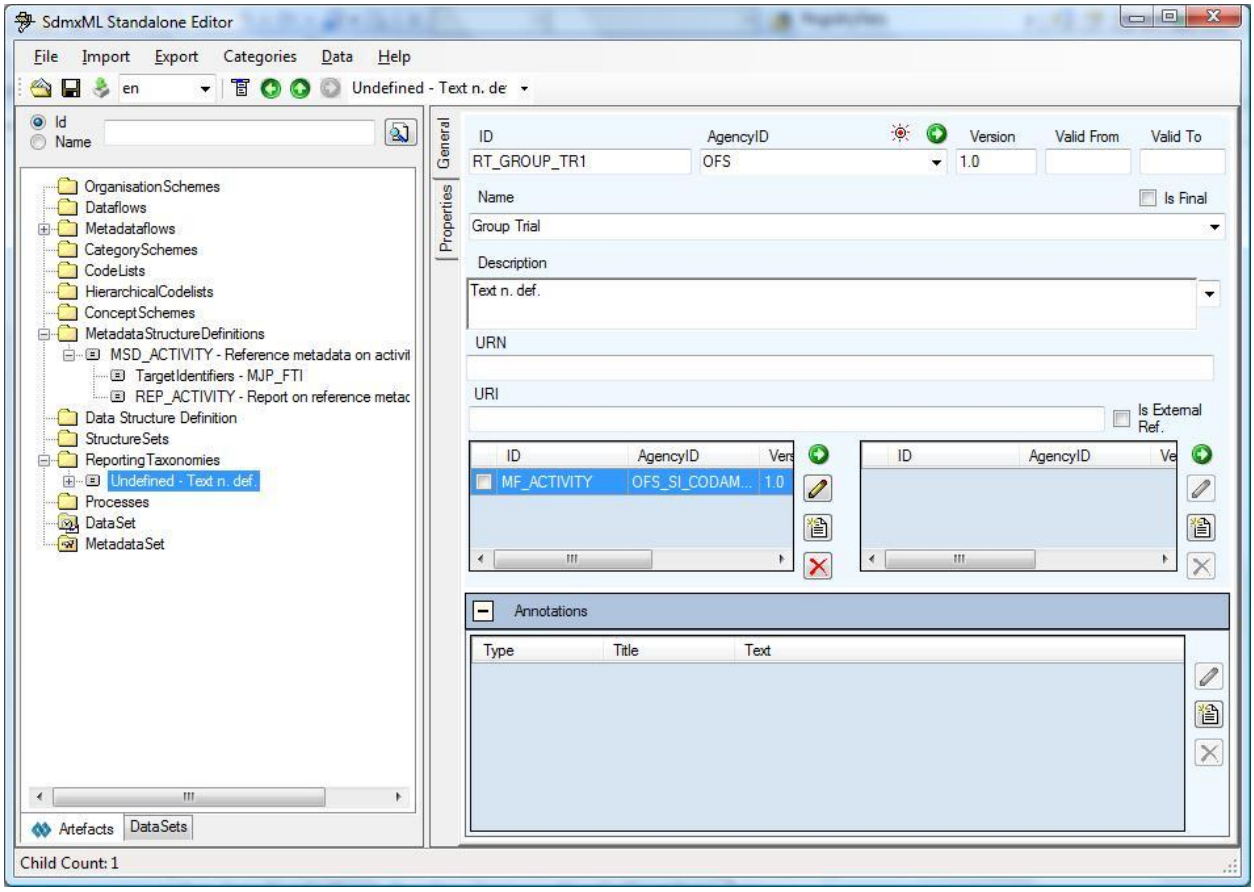
The Application can only work on artefact attributes and on references. The categories Tab support is not implemented.



7.10 Reporting Taxonomy

7.10.1 Artefact detail

Reporting Taxonomy is very similar to category scheme artefact except that it holds data and metadata flows references in the artefact itself.

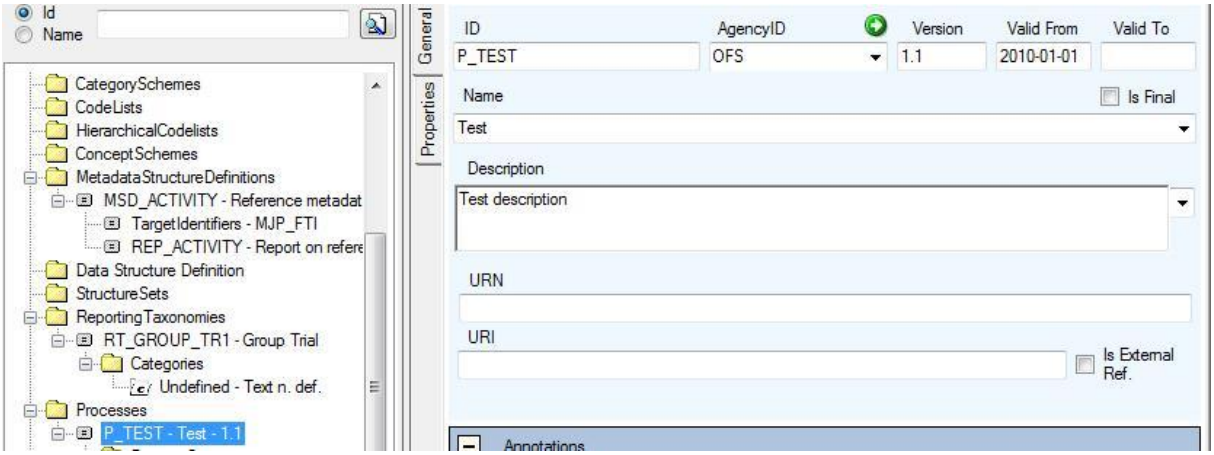


Please refer to Category schemes documentation for the categories management in this artefact.

7.11 Processes

7.11.1 Artefact detail

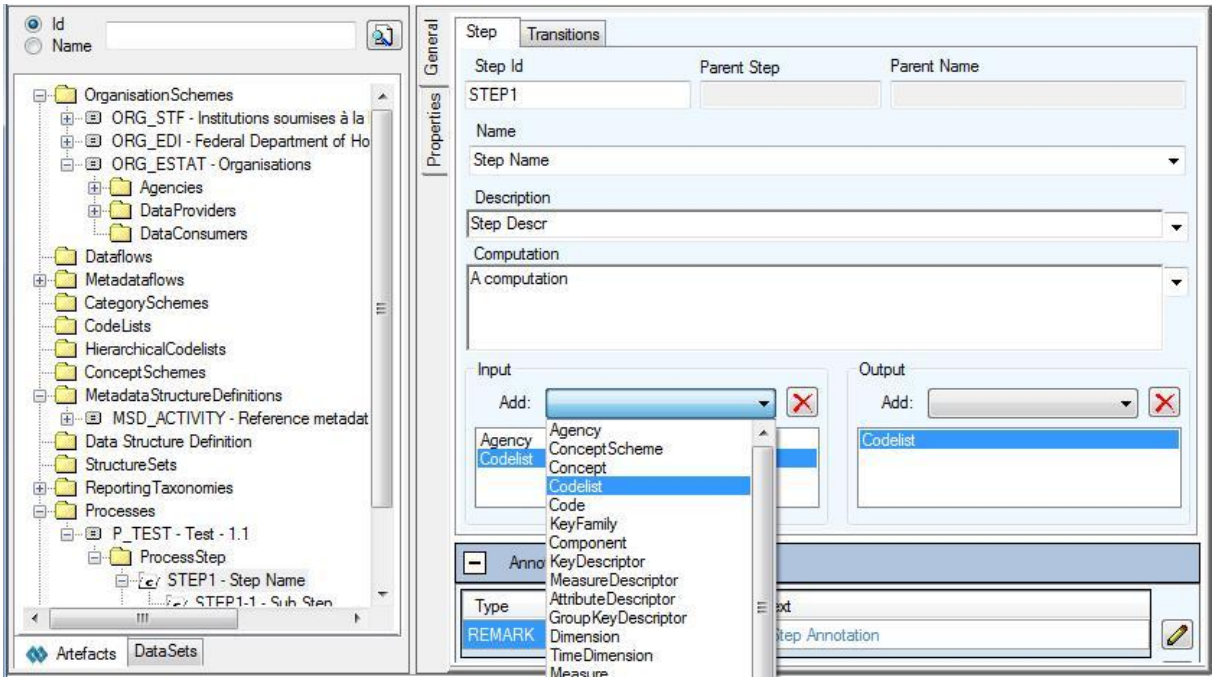
The detail screen manages the standard artefact attributes.
Name and Description support multi language edition.



7.11.2 Process Step detail

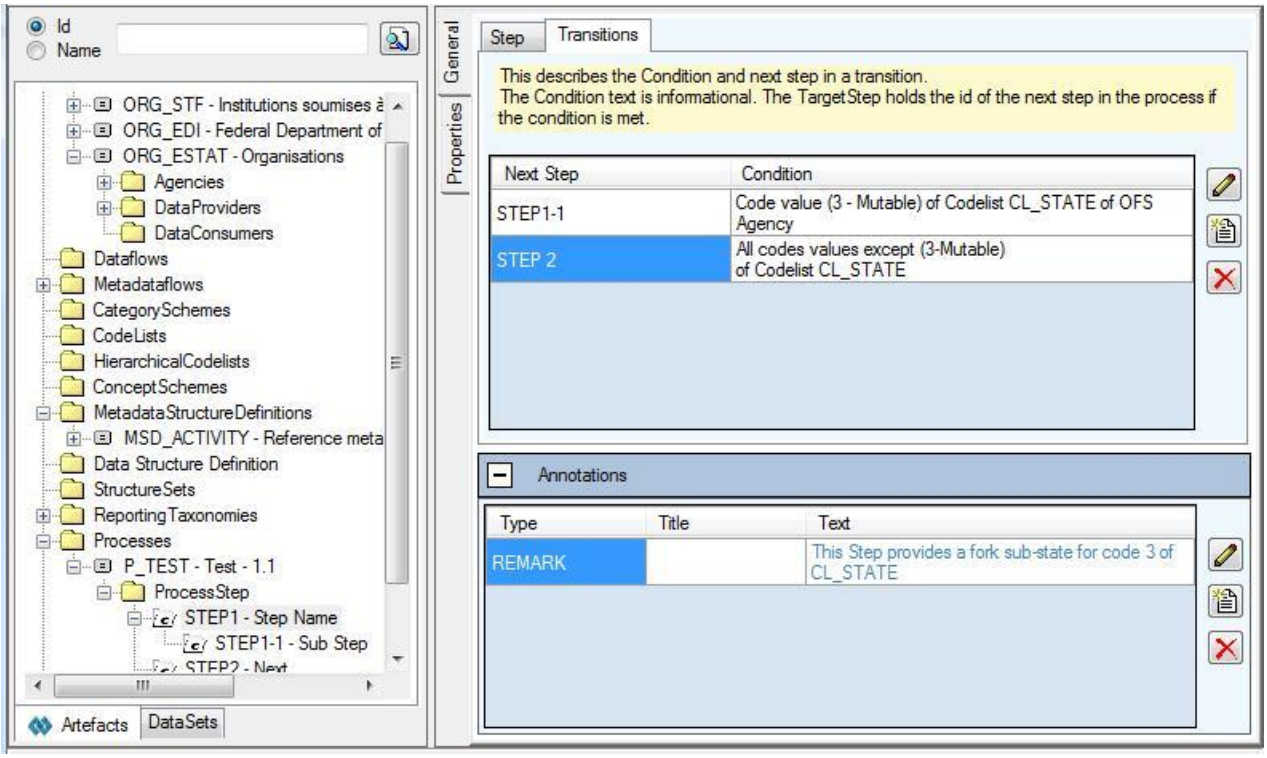
The Process steps can be hierarchic if required. Steps are mainly descriptive. At each step it can be given a name, a description and Computation text information.

It is also possible to refer input and output artefact type objects that are concerned by the step..



7.11.3 Process Step Transitions

Transition to next step can be documented with a list of next step ID and a condition documentation internationalized text.



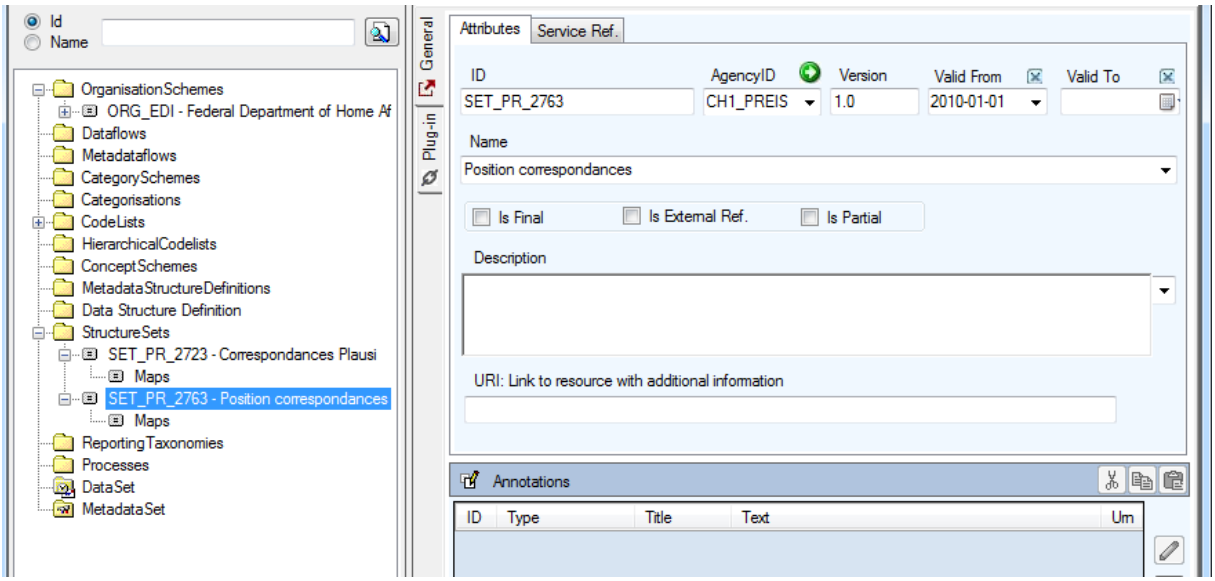
7.11.4 Functionalities

	Edit selected transition in a specialized internationalised dialog box.
	Creates a new transition attached to selected step item.
	Deletes selected transition.

7.12 Structure Sets

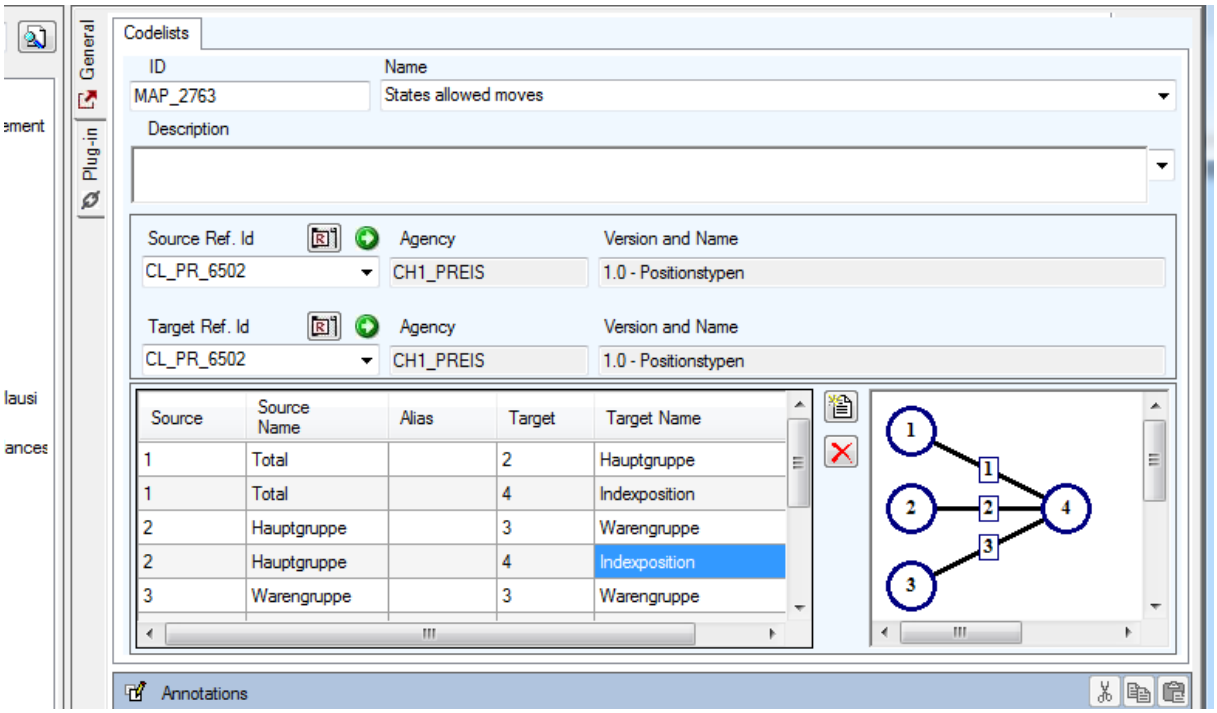
7.12.1 Artefact detail

The detail screen manages the standard artefact attributes.
Name and Description support multi language edition.



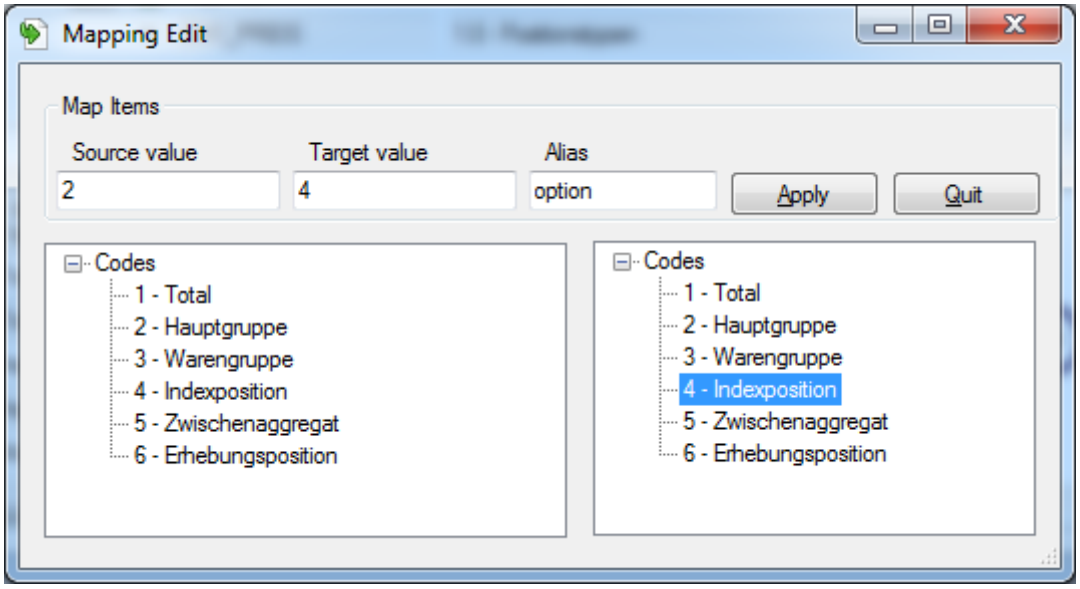
7.12.2 Maps detail

Currently only Map for code list correspondences is implemented. The source and target code list are specified first. Then the link between codes can be specified through a specialized dialog box.



A graphical view presents the mapped items corresponding to the currently selected value in map grid.

	Creates a set of mappings according to availables codes.
	Deletes selected map item.



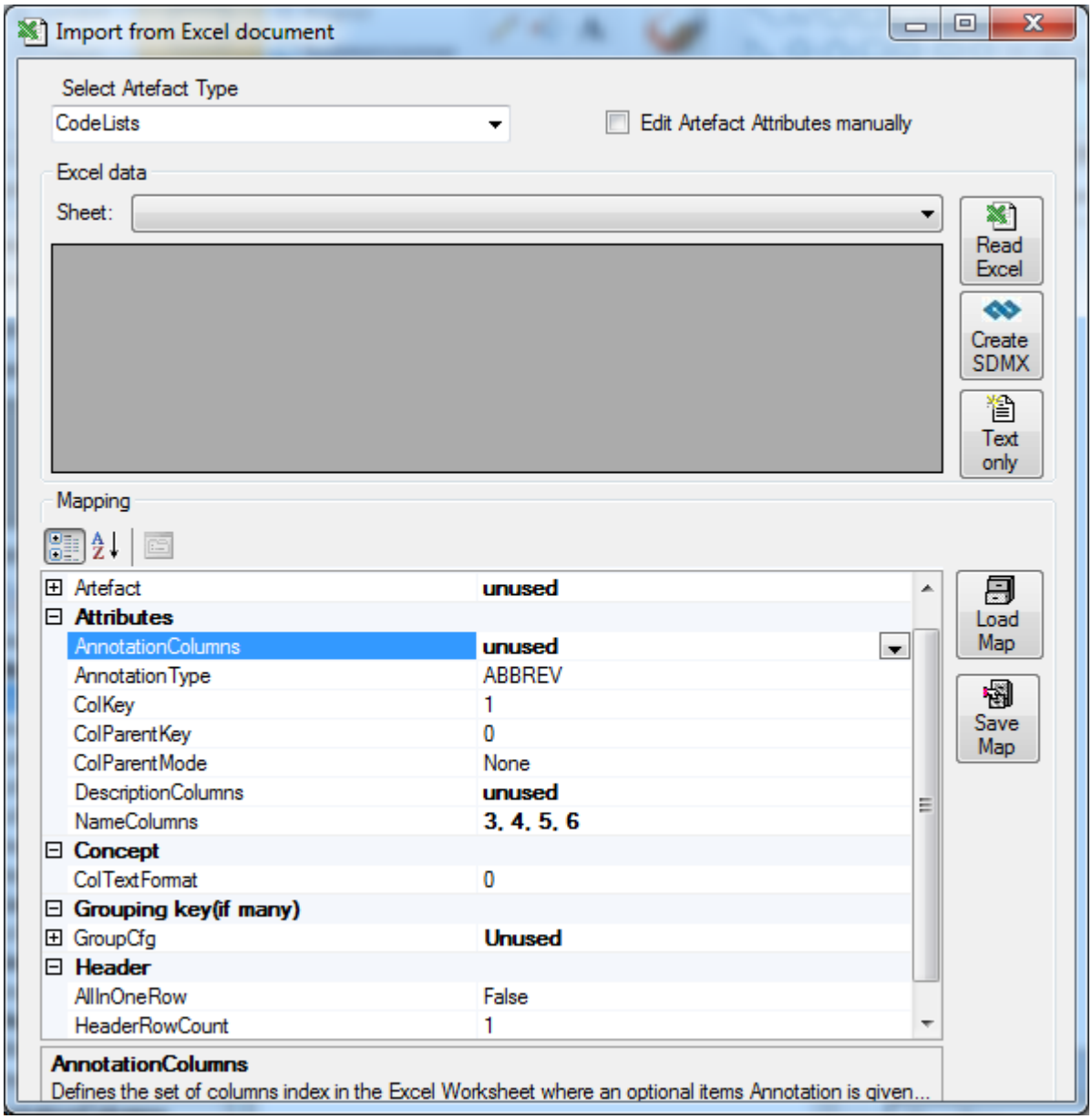
The mapping dialog box allows user to specify source and target items bs typing values in corresponding text boxes or by selecting value in codes trees. The apply button record each mapping. To terminate mapping creation the user activates the “quit” button. The set of mapping is reflected in the grid of the map detail view.

The dialog box checks the submitted mapping and refuses it if it is already defined in the map.

8 Excel

The application supports import of Excel document for artefacts like: code list, hierarchical code lists, concepts and categories. A simple mapping can be defined between excel worksheet columns and artefact attributes.

8.1 Excel Import dialog



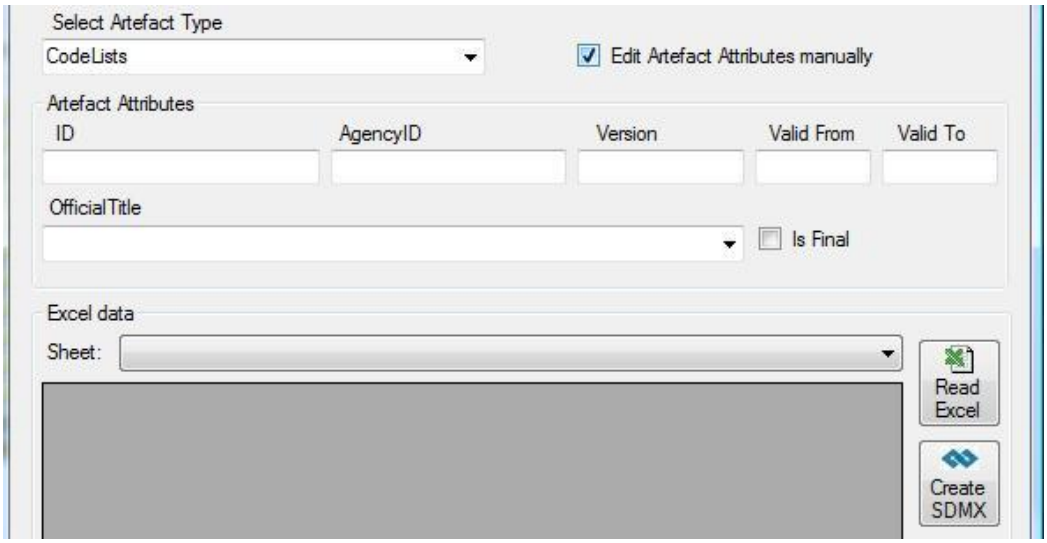
The dialog allows selecting the Excel document and work sheet to use. The mapping between rows and columns of the Excel sheet and of the artefact attributes can be defined in the Mapping property editor. This mapping can be saved in a file in order to reuse it later.

The created sdmx artefact is loaded on the application tree. It is also possible to import only updated text, for example for others languages, if a same artefact is already loaded in main application tree and the “Text only” button is used in import dialog. This function allows also importing annotations of different types one by one.

The dialog is a top window and it is possible to work in application when it is raised.

If only the items are part of the excel sheet, the artefacts attributes can be edited manually by checking the following checkbox:

The Dialog box displays an edition area for artefact attributes:



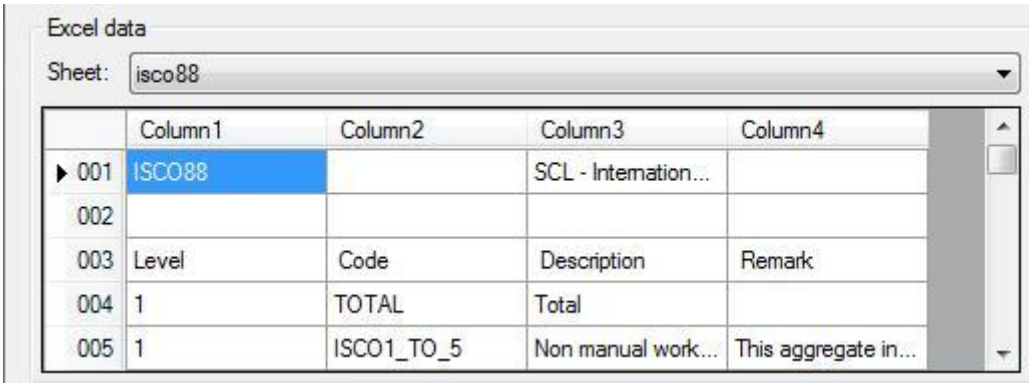
8.2 Loading an Excel Worksheet

This function opens a “file open” dialog box. When the excel document is selected and loaded, The sheet drop down is filled with the names of all excel worksheets.

The Grid displays the content of the first excel worksheet (see below). It is possible to select another worksheet with the drop down box. As soon as another sheet is selected, the grid is refreshed with the sheet content. Rows and columns are numbered to help mapping process.

The displayed cells can be edited if a small modification is to be done.

The application uses the free component “ExcelDataReader” of Alex Messir.


Excel data
Sheet: isco88
Column1 Column2 Column3 Column4
001 ISCO88 SCL - Internation...
002
003 Level Code Description Remark
004 1 TOTAL Total
005 1 ISCO1_TO_5 Non manual work... This aggregate in...

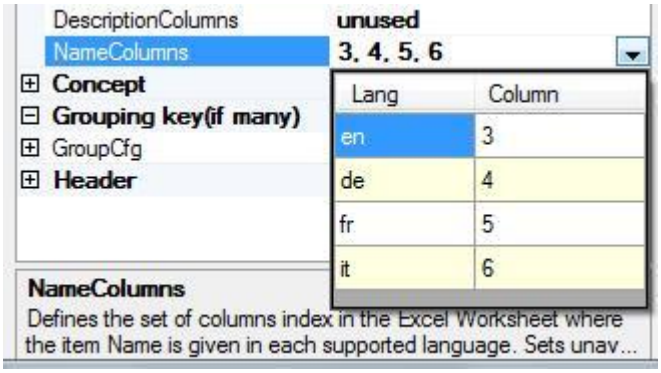
8.3 Mapping property editor

The mapping properties are as follow.

8.3.1 Internationalized text mapping

For the texts that are defined in some of the supported languages of the application, a column mapping must be given for each language. The property editor displays a dropdown grid for specifying column index for languages to import. For unused language, sets the column index to zero.

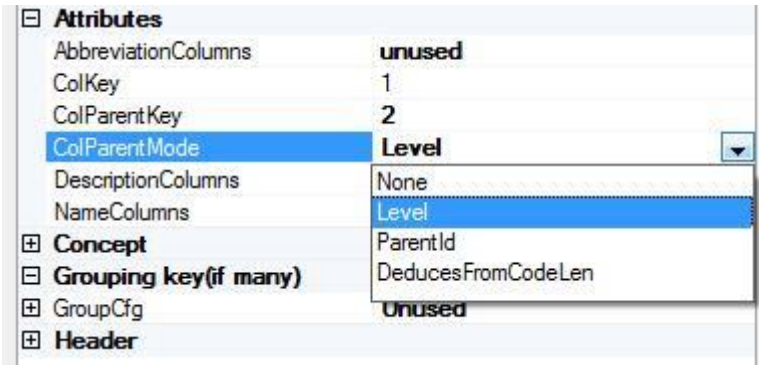
If all columns are set to zero, the property displays “unused”. Note that a least one language for name column is required by SDMX specifications.



At the bottom of the property editor, an information area gives explanation about the selected property.

8.3.2 Hierarchy mapping

If requested, the hierarchy can be inserted in the SDMX artefact. The hierarchy can be specified according to different methods.



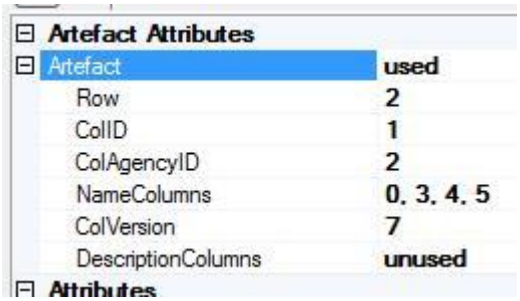
- ColKey specifies the column of the item ID (code). It is mandatory.
- ColParentKey specifies the column of the parent information. This is optional or mandatory for some modes of the hierarchy.
- ColParentMode specifies the mode to use for determining the hierarchy.
    - None: hierarchy unused
    - Level: Hierarchy constructed from item code level number. In this case the *ColParentKey* information must indicate the level column.
    - ParentId: Hierarchy constructed from item code parent information. In this case the *ColParentKey* information must indicate the parent id column.
    - Deduces from code length: The hierarchy id deduced from code size changes. In this case *ColParentKey* is not required.

8.3.3 Header rows

This information indicates the number of rows to skip in the excel sheet before importing items codes and texts.

8.3.4 Artefact main attributes

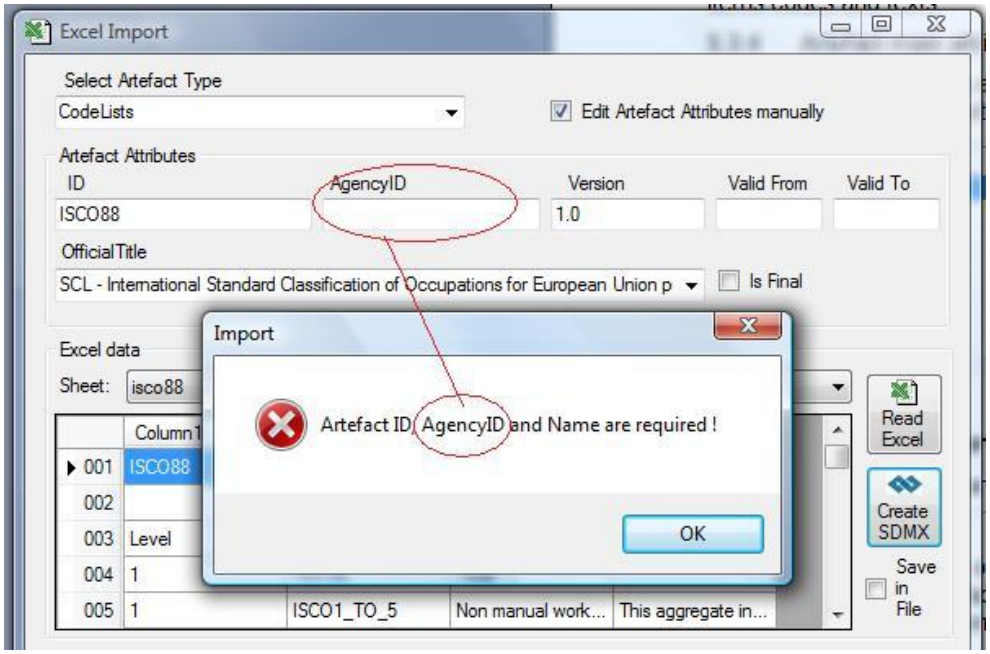
If artefact main attributes are placed in one horizontal row of the excel sheet they can also be imported during the import process. If not, uses the manual edition check box.



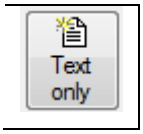
8.4 Artefact creation

8.4.1 Function create

This function operates the conversion of the Excel worksheet content to the specified artefact type according to the specified mapping. If some of the required artefact attributes are missing, the application opens the manual edit box and raises a warning message. Simply fills missing info and reactivates function.



8.4.2 Function update text



This function checks that the same artefact is present in main tree memory and it updates items texts and annotations content.

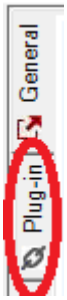
8.4.3 Samples

05 Nomen Langue.xls map05NomenLangue.map	This sample file shows how to import a simple code list that is internationalized. The associated mapping file maps German, French and Italian. Note that Artefact ID and Agency must be written manually.
Eurostat_CODELIST_Sample.xls mapIsco88.map	This sample file shows how to import a hierarchy's artefact using levels. The Isco88 is located on the second worksheet of the excel document. Note that Artefact Agency must be written manually.

8.5 Excel Export

Selected artefact can be exported as an **Excel XML** document. The first worksheet contains the information about the artefact. The next worksheet(s) contain the detail of artefact content included annotations.

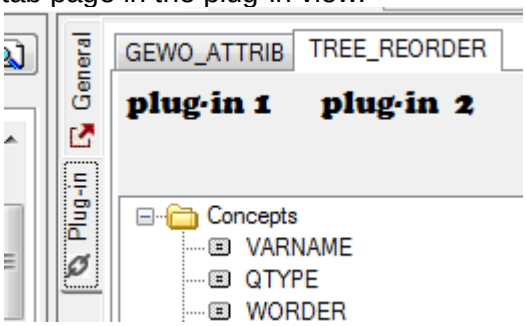
Note: to be able to use the document for an import (after some corrections) it must be saved as an excel document type (Excel 2003 workbook)



9 Plug-in

The application supports the addition of plug-in. These components should be written in MS.NET and implements the *IPlugin* interface provided by SAE. The class to instantiate must derive from a window form user control. The plug-in can exchange information to main program. The information has to be an SDMX-ML XML Document.

Plug-in are loaded under the associated tab in the detail view. Each instantiated plug-in matches a tab page in the plug-in view.



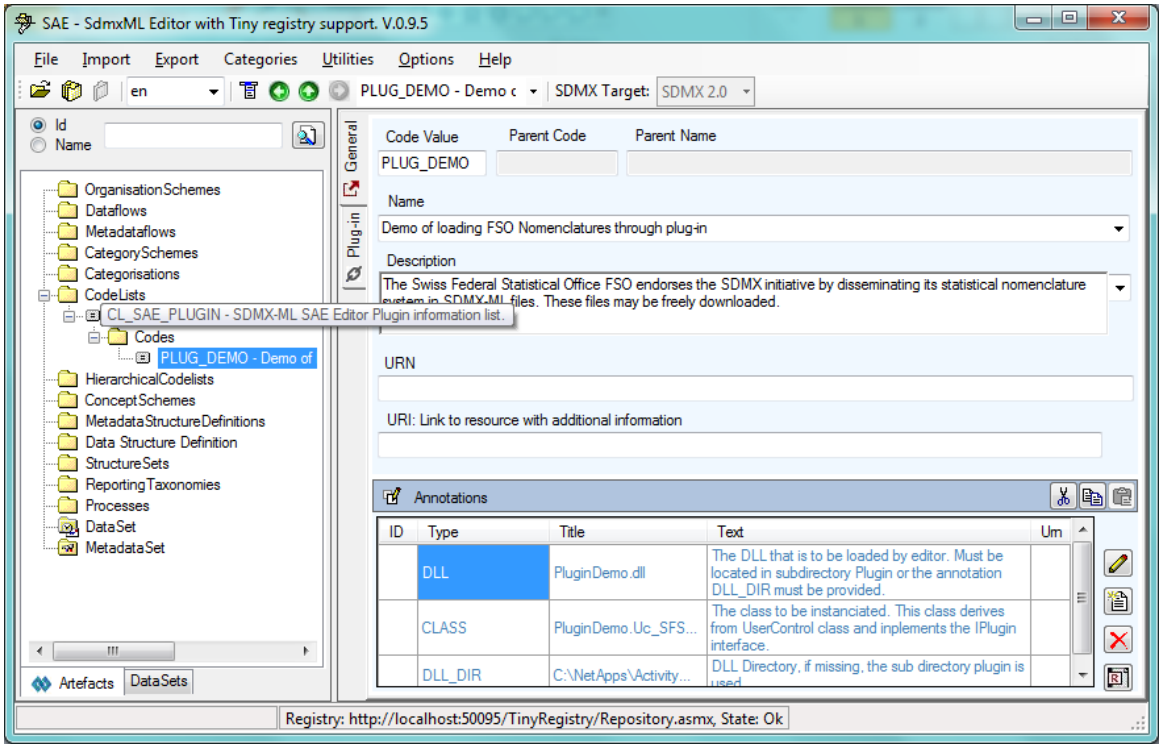
9.1 Plug-in installation

The installation instructions are given in a sdmx codelist document, this document is named *"CL_SAE_PLUGIN.xml*" and must be placed under the sub directory "*Plugin*" of the editor.

Each plug-in to install is described as one code of the code list. For each code, two annotations are mandatory and one is optional. The code id is used for form tab naming. Name and description are used to document the plug-in.

9.1.1 Required annotations

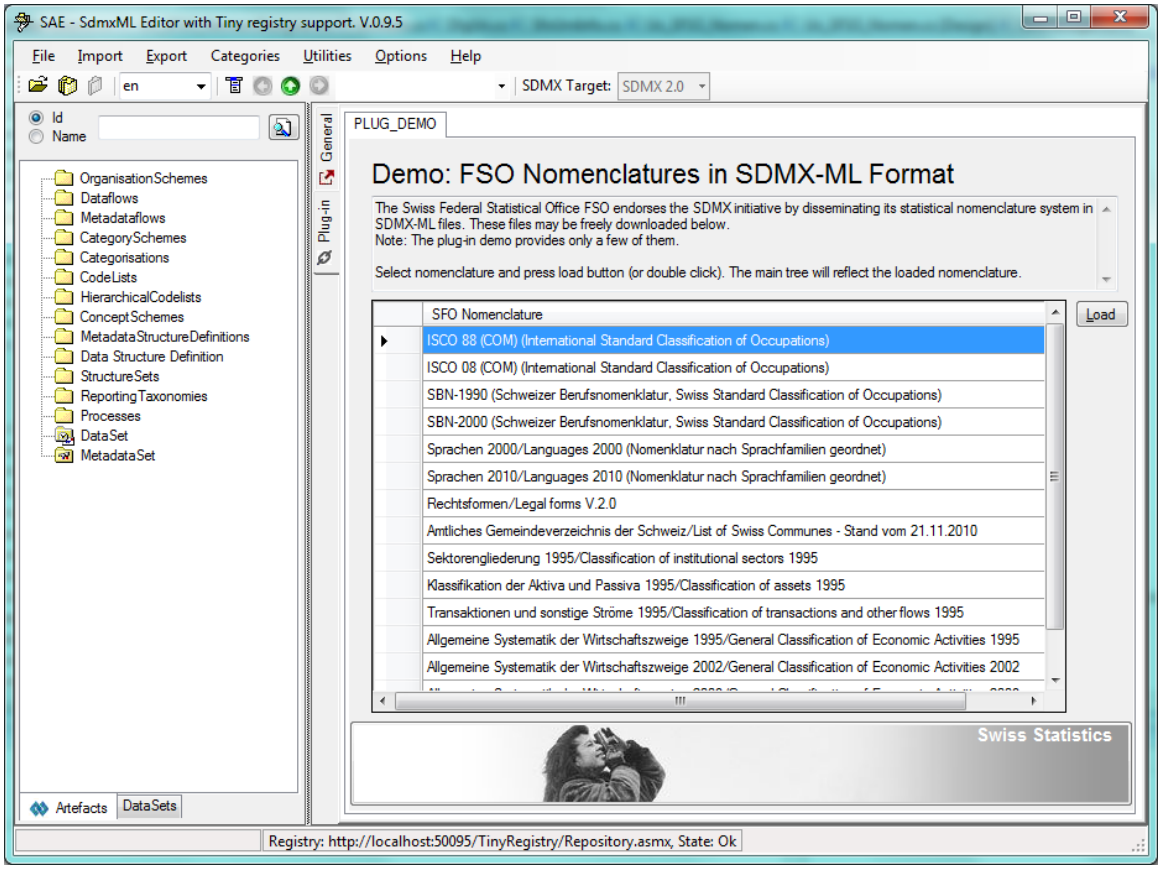
Ann. Type	Ann. Title	Description
DLL	Plugindll.dll	Holds the name of the DLL that is to be loaded by editor. This library must be located in subdirectory '*Plugin'* or the annotation DLL_DIR has to be provided for specifying the dll directory.
CLASS	Namespace.classname	Holds the name of the class to be instantiated. This class derives from *UserControl* class and implements the *IPlugin* interface.
DLL_DIR	FullDirectory	Optional. Indicates the directory to use for locating the DL, if not provided in sub directory *Plugin*.


SAE - SdmxML Editor with Tiny registry support. V.0.9.5
File Import Export Categories Utilities Options Help
SDMX Target: SDMX 2.0
Code Value
Parent Code
Parent Name
PLUG_DEMO
Name
Demo of loading FSO Nomenclatures through plug-in
Description
The Swiss Federal Statistical Office FSO endorses the SDMX initiative by disseminating its statistical nomenclature
CL_SAE_PLUGIN - SDMX-ML SAE Editor Plugin information list.
URN
URI: Link to resource with additional information
Annotations
ID Type Title Text Um
DLL PluginDemo.dll The DLL that is to be loaded by editor. Must be located in subdirectory Plugin or the annotation DLL_DIR must be provided.
CLASS PluginDemo.Uc_SFS... The class to be instanciated. This class derives from UserControl class and inplements the IPlugin interface.
DLL_DIR C:\NetApps\Activity... DLL Directory, if missing, the sub directory plugin is
Artefacts DataSets
Registry: http://localhost:50095/TinyRegistry/Repository.asmx, State: Ok

The above screen presents a plug-in instructions sample.

9.2 Plug-in demo

A demonstration plug-in is delivered with the editor. It illustrates the event mechanism used to transmit SDMX-ML documents to the main application. The corresponding source code is located under *PluginDemoSource* sub-directory.


SAE - SdmxML Editor with Tiny registry support. V.0.9.5
File Import Export Categories Utilities Options Help
SDMX Target: SDMX 2.0
PLUG_DEMO
Demo: FSO Nomenclatures in SDMX-ML Format
The Swiss Federal Statistical Office FSO endorses the SDMX initiative by disseminating its statistical nomenclature system in SDMX-ML files. These files may be freely downloaded below.
Note: The plug-in demo provides only a few of them.
Select nomenclature and press load button (or double click). The main tree will reflect the loaded nomenclature.
SFO Nomenclature
Load
ISCO 88 (COM) (International Standard Classification of Occupations)
ISCO 08 (COM) (International Standard Classification of Occupations)
SBN-1990 (Schweizer Berufsnomenklatur, Swiss Standard Classification of Occupations)
SBN-2000 (Schweizer Berufsnomenklatur, Swiss Standard Classification of Occupations)
Sprachen 2000/Languages 2000 (Nomenklatur nach Sprachfamilien geordnet)
Sprachen 2010/Languages 2010 (Nomenklatur nach Sprachfamilien geordnet)
Rechtsformen/Legal forms V.2.0
Amtliches Gemeindeverzeichnis der Schweiz/List of Swiss Communes - Stand vom 21.11.2010
Sektorengliederung 1995/Classification of institutional sectors 1995
Klassifikation der Aktiva und Passiva 1995/Classification of assets 1995
Transaktionen und sonstige Ströme 1995/Classification of transactions and other flows 1995
Allgemeine Systematik der Wirtschaftszweige 1995/General Classification of Economic Activities 1995
Allgemeine Systematik der Wirtschaftszweige 2002/General Classification of Economic Activities 2002
Swiss Statistics
Artefacts DataSets
Registry: http://localhost:50095/TinyRegistry/Repository.asmx, State: Ok

9.3 Interface definition

```
namespace PluginInterface
{
    /// <summary>
    /// Contract for the implementation of a minimal set of methods by the plugins.
    /// These methods are used by the main (incorporating) application to find and
    /// recognize the plugins. This minimal set of method is achieved by making this
    /// interface, which declares the methods, that are to be implemented by the plugins.
    /// </summary>
    public interface IPlugin
    {
        /// <summary>
        /// An event generator to notify a SDMX-ML document is to be consumed by the main
       /// application.
        /// </summary>
        event PluginEvent PluginEventExposed;

        /// <summary>
        /// this method will be called when the plugin is activated if 'RequiresInit' is set
        /// to true.  </summary>
        /// <param name="xDoc">null or an XML SDMX-ML document representing the artefact
        /// selected in the main application.
        /// </param> <param name="obExtend">Reserved</param>
        void Init(XmlDocument xDoc, object obExtend);

        /// <summary>
        ///  VisualComponent allows the main application to retrieve the Visual control of the
        ///  plugin and to incorporate it in the reserved area or as a dislog box.
        /// </summary>
        /// <returns>the user control implementing this interface</returns>
        UserControl VisualComponent();

        // If true, the Init method of the plugin is called when activated.
        // if false, the Init method is not called.
        bool RequiresInit { get; }

    }


    //The event architecture offers a mechanism for an event generator to notify an event
    // consumer that something happened. The event generator does not need to know anything
    // about the consumer. Instead it publishes a method definition in the form of a delegate,
    // plus an event handle. When the event happens, the generator will call the unknown event
    // handling method of the consumer through the event handle.
    public class PluginEventArgs : EventArgs
    {
        public XmlDocument DocSdmxMl { get; set; }

        public PluginEventArgs(XmlDocument xDoc)
        {
            DocSdmxMl = xDoc;
        }
    }

    /// <summary> Publishes the method definition in the form of a delegate </summary>
    /// <param name="sender"></param>/ <param name="e">PluginEventArgs instance</param>
    public delegate void PluginEvent(object sender, PluginEventArgs e);
```